Presented to:

AMS Seminar Series

NASA Ames Research Center

April, 19, 2016

Solution Algorithm On Unstructured Grids Using Hamiltonian Paths and Strand Grids

IAW DoD Directive 5230.24, insert appropriate distribution statement

TECHNOLOGY DRIVEN. WARFIGHTER FOCUSED.

Presented by:

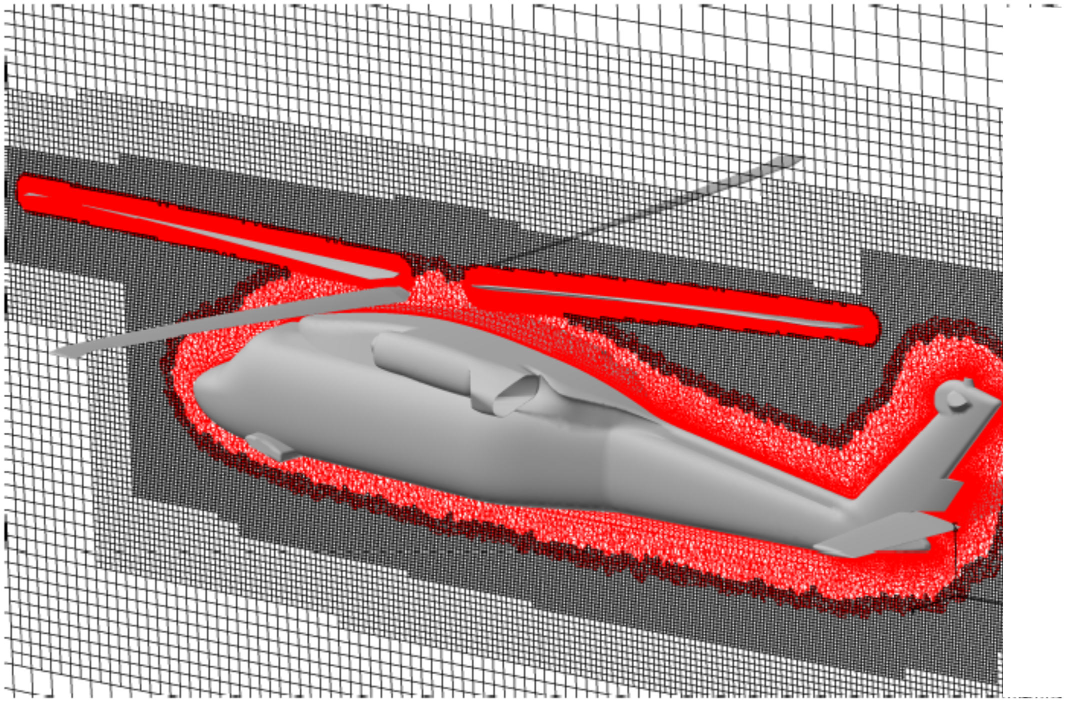
Jayanarayanan Sitaraman

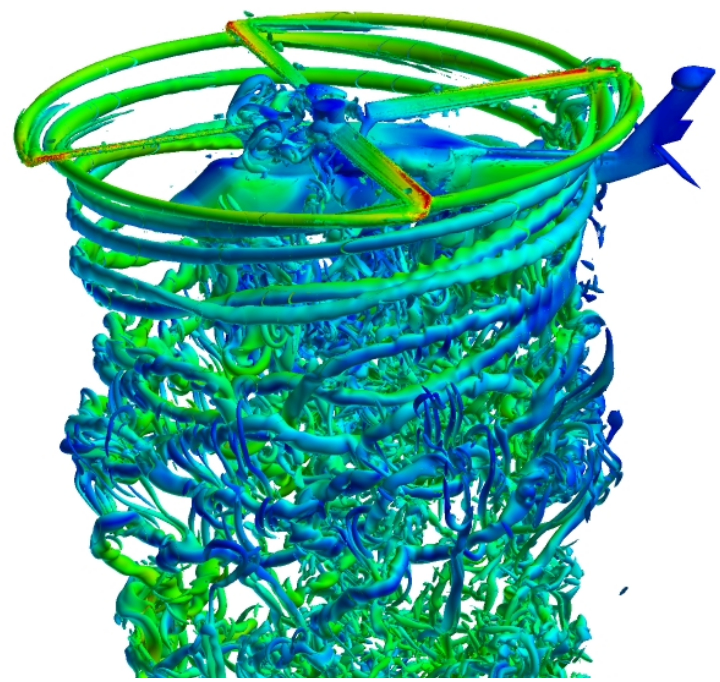
Senior Scientist

Parallel Geometric Algorithms LLC

Sunnyvale, CA

Introduction





U.S Army HELIOS framework

- **Multiple mesh system using overset technique**
  - **Unstructured/ structured or strand grids for near-body region**
  - **Structured Cartesian grids for off-body region (high-order discretization)**
- **Limitation of unstructured solver still remain in the near-body region**
  - **Limited accuracy (2nd order FV type discretization schemes)**
  - **3 to 10 times slower than corresponding structured grid solvers**

Motivation(1)

- Structured grid solution methods are more well established and provided efficient solution methods
  - Stencil based discretization
  - Approximate factorization of the implicit operator and line-implicit solution schemes
  - "High-order type" numerical schemes are mature

- Unstructured grids provide versatility to model complex geometry
  - Generally quite a bit slower than corresponding structured grid solvers
  - High order quite difficult and expensive within finite volume framework
    - Compute gradients, gradients of gradients, limiters, flux correction operators etc
  - Finite-Element based high-order is still maturing

Motivation(2)

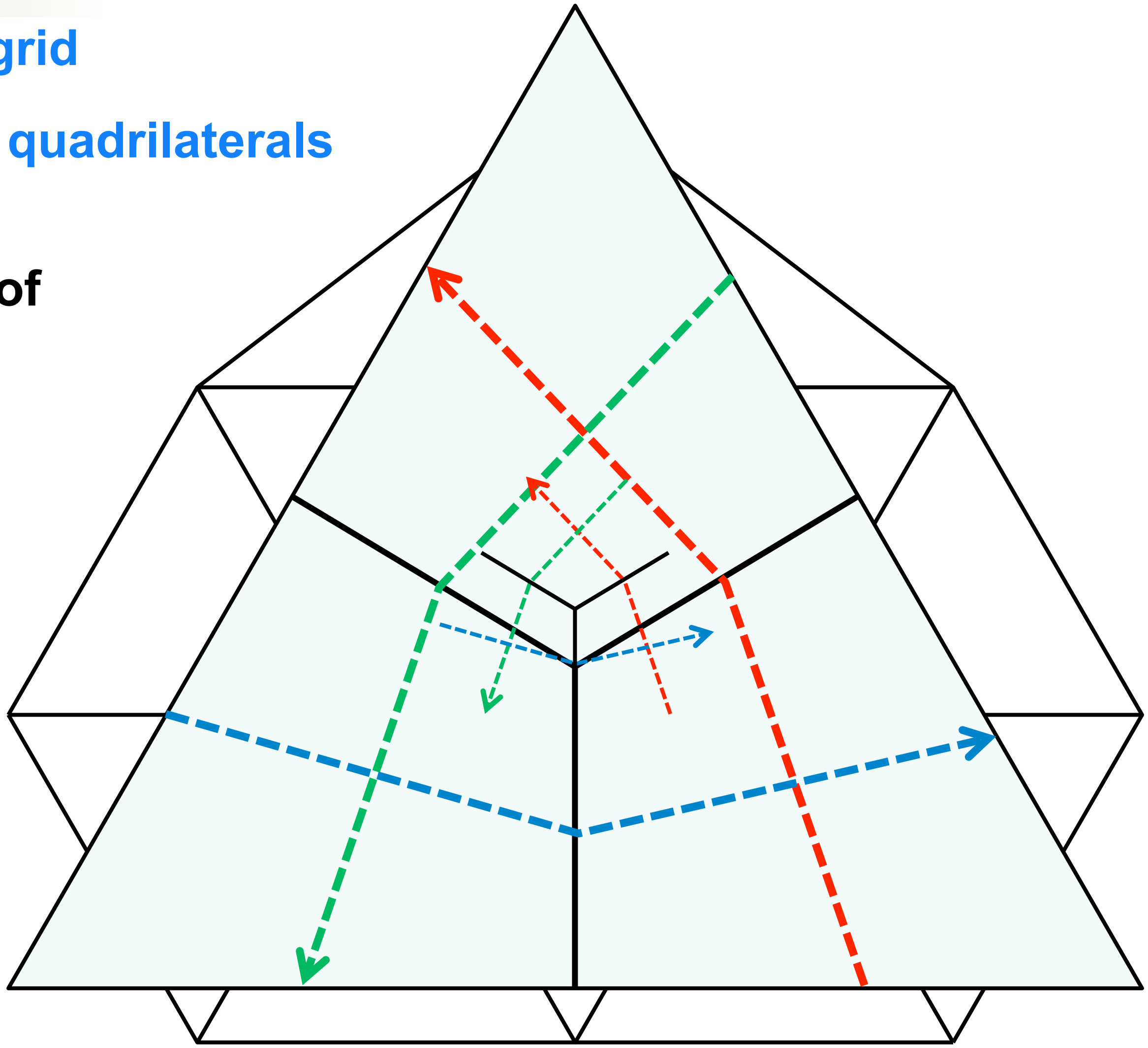
- Can we find structure in unstructured grids such that line-implicit solutions and stencil-based discretization can be used?
  - Early work by Martins and Lohner (1993) and Hassan et al. (1989)
  - Abandoned because
    - difficulty in finding lines in pure unstructured grids (NP-hard problem)
    - Difficulty in achieving nesting of lines
  - Line-implicit inversion in the wall-normal direction (prizm layers) relieves stiffness caused by stretching (Mavriplis (1997))

- Can we find a method that can easily and always locate lines and also provide the required nesting?
  - Yes.. If you divide the triangles into quadrilaterals (2-D)
  - Yes.. If surface lines can be combined with strands (3-D)

- This work is somewhat off the beaten path and inspired by toys ☺

Subdivision and Hamiltonian Paths

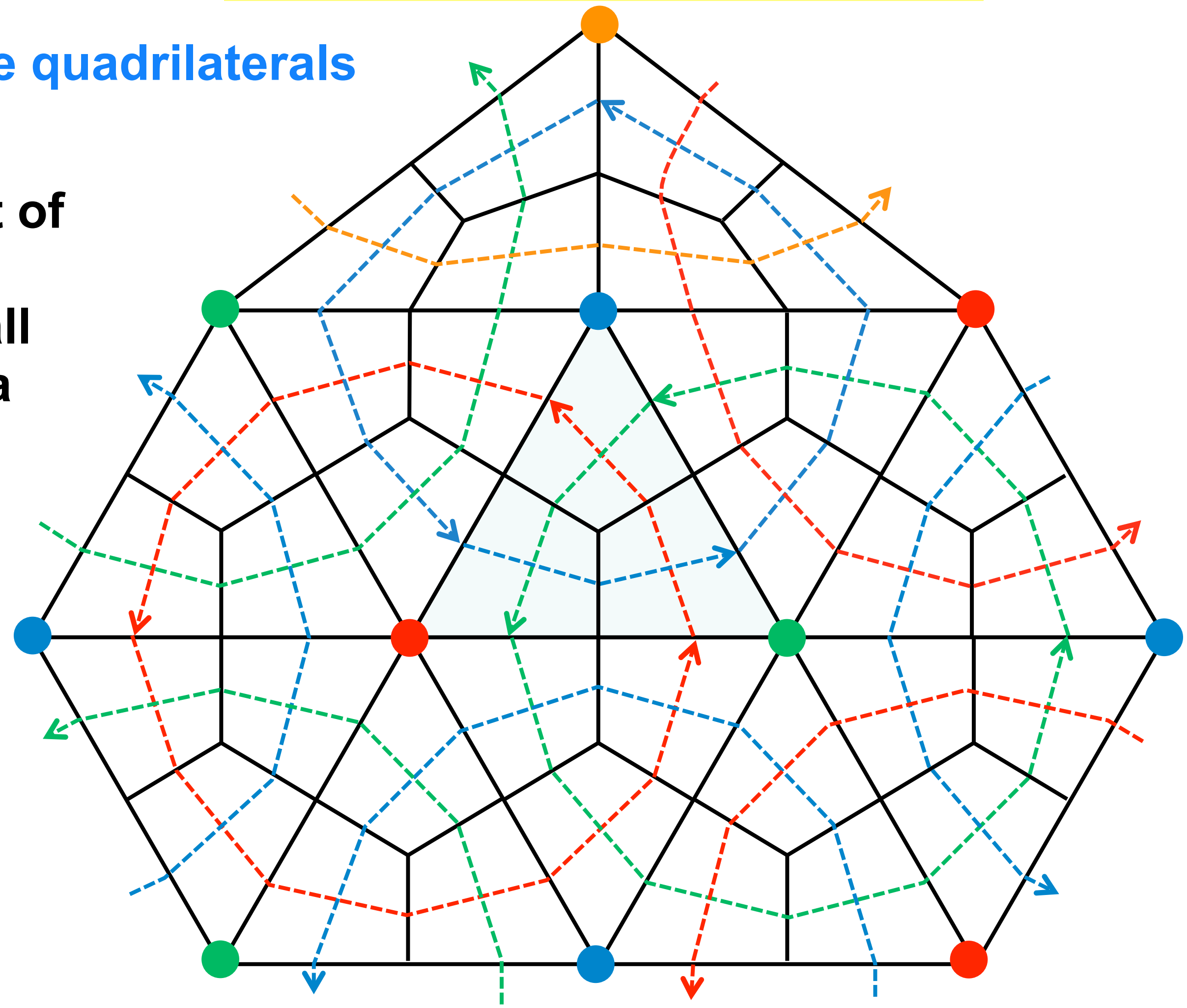
- Triangular unstructured grid
- Divide triangle into three quadrilaterals
- Loops constructed by connecting the midpoint of edges



Subdivision and Hamiltonian Paths

- Triangular unstructured grid
- Divide triangle into three quadrilaterals
- Loops constructed by connecting the midpoint of edges
- Loops formed through all triangles connected by a triangular node
- Each face part of only one distinct loop
- Each cell centroid is intersected by loops of different colors

The different coloured loops are the Hamiltonian paths

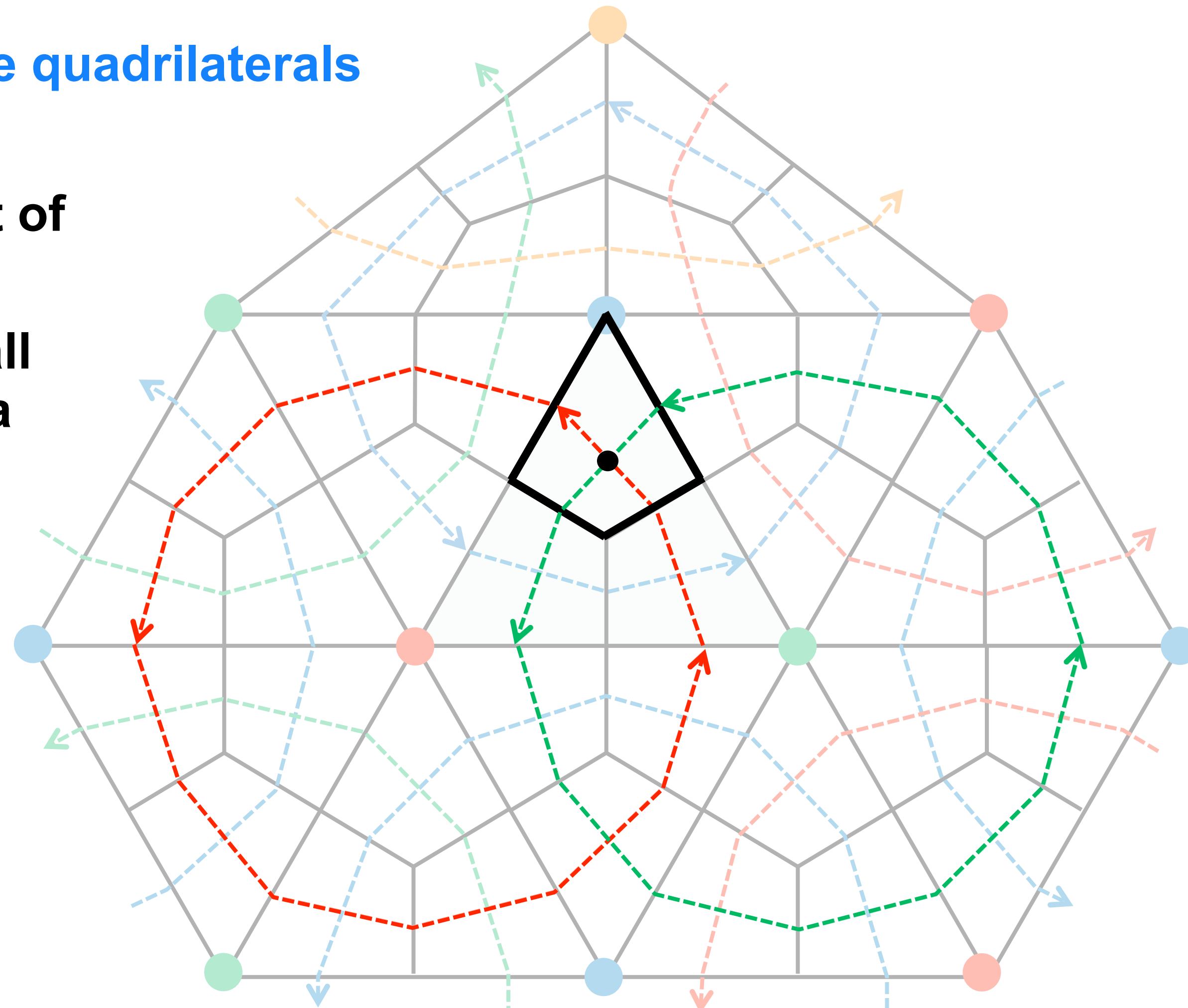


UNCLASSIFIED

Subdivision and Hamiltonian Paths

- Triangular unstructured grid
- Divide triangle into three quadrilaterals
- Loops constructed by connecting the midpoint of edges
- Loops formed through all triangles connected by a triangular node
- Each face part of only one distinct loop
- Each cell centroid is intersected by loops of different colors

The different coloured loops are the Hamiltonian paths

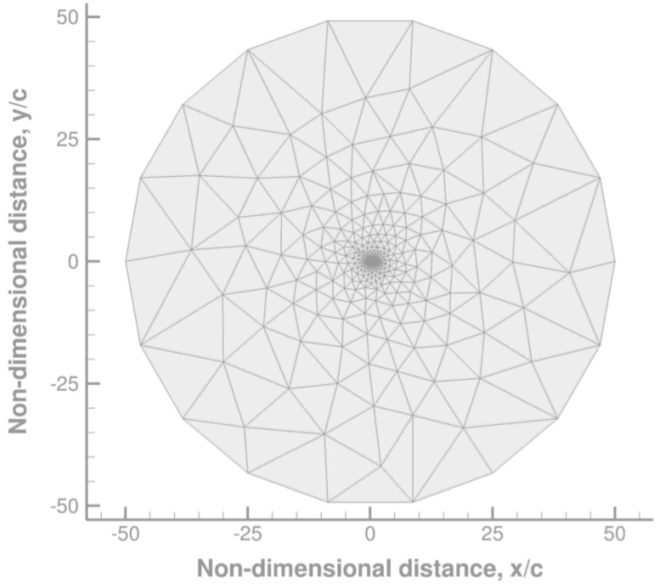


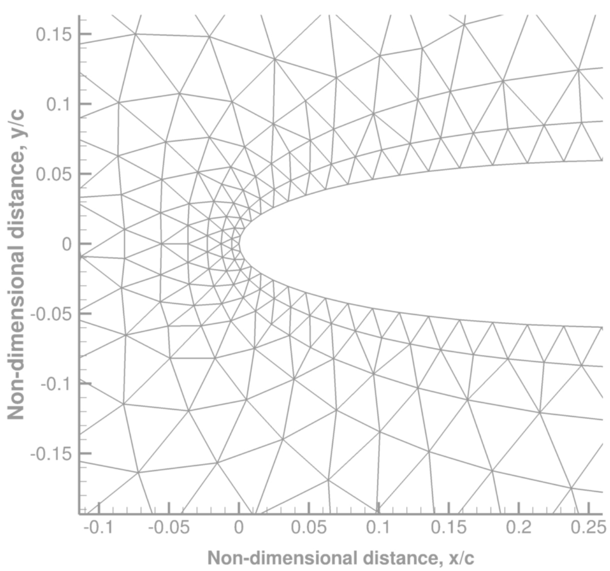
UNCLASSIFIED

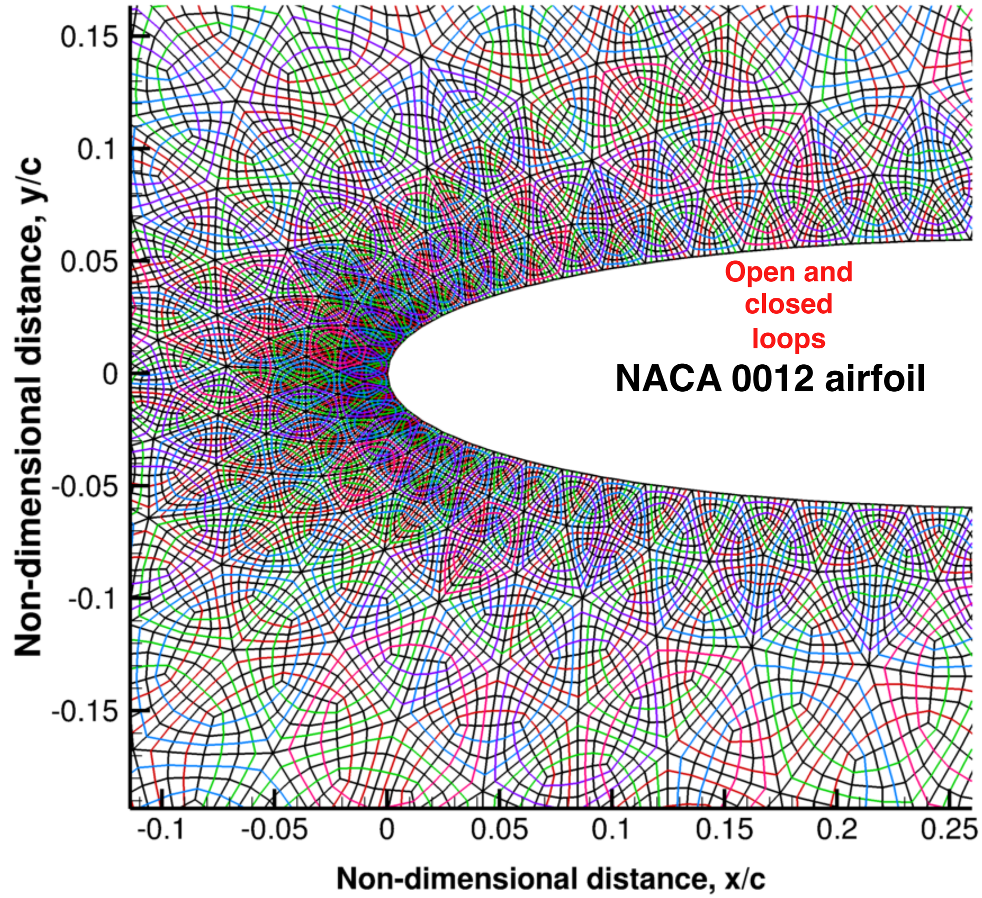
Chronology

- Work started as an unfunded hobby project at UW
  - Scitech 2014 paper on 2-D work
  - CREATE A/V management got interested and wanted to fund a student/post-doc to extend the concept to 3-D as a possible near-body solver for HELIOS, but I left UW in 2014
- Prof. Jim Baeder's group at Univ of MD had couple of students who were really interested and they secured a PETT grant from CREATE to continue this work.
  - Most of the 3-D work shown here was performed by Bharath Govindarajan, Yong Su Jung and Jim Baeder at UMD
- Papers presented
  - Scitech 2014 (Sitaraman and Roget)
  - AHS 2015 (Govindarajan et al.)
  - Scitech 2016 (Jung et al.)
  - AHS 2016 (Govindarajan et al. to be presented)
  - JCP article (Govindarajan et al. pending)

Hamiltonian Paths on an Airfoil Grid







- Hamiltonian loops now provide structure!
  - Loops are equivalent to traditional lines
- Take advantage of “lines” on a purely unstructured grid

Governing equations

- **2-D Compressible Navier-Stokes (Laminar)**

$$\frac{\partial q}{\partial t} + \nabla \cdot [F(q), G(q)] = 0$$

$$\frac{q^{n+1} - q^n}{\Delta t} = - \sum_{i \in nfaces} \hat{F}_i(q^{n+1}) \cdot ds_i$$

→ **Steady state formulation**

$$\left[\frac{I}{\Delta t} + \frac{\partial R}{\partial q} \right] \Delta q = -R(q^n)$$

→ **Standard Newton Linearization**

$$\hat{F} = \frac{1}{2}((F(q_L, d\vec{S}) + F(q_R, d\vec{S})) - |A(q_L, q_R, d\vec{S})|(q_R - q_L))$$

→ **Roe's approximate Rieman Solver for inviscid flux**

Viscous terms are computed using 2nd order central finite differencing

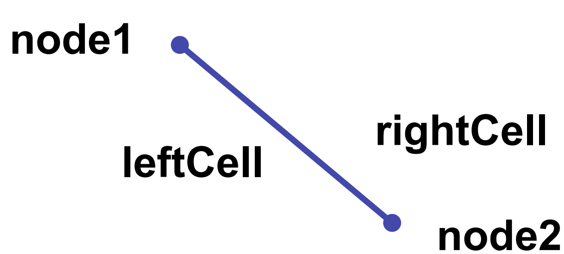
Solver Data Structures for 2-D

$nnodes$	total number of vertices
$ncells$	total number of cells
$nfaces$	total number of faces
$ncolors$	number of colors
$nchains$	number of Hamiltonian paths
$chainsPerColor[]$	number of paths in each color
$faceStartPerChain[]$	starting face index for each path
$chainConn[]$	connectivity of chains, e.g. $chain_i$ has $face \in [chainConn[f_i], chainConn[f_{i+1} - 1]]$
$faces[]$	list of faces (6 ints per face in 2D)
$q[]$	field variables $[4 \times ncells]$

Quads

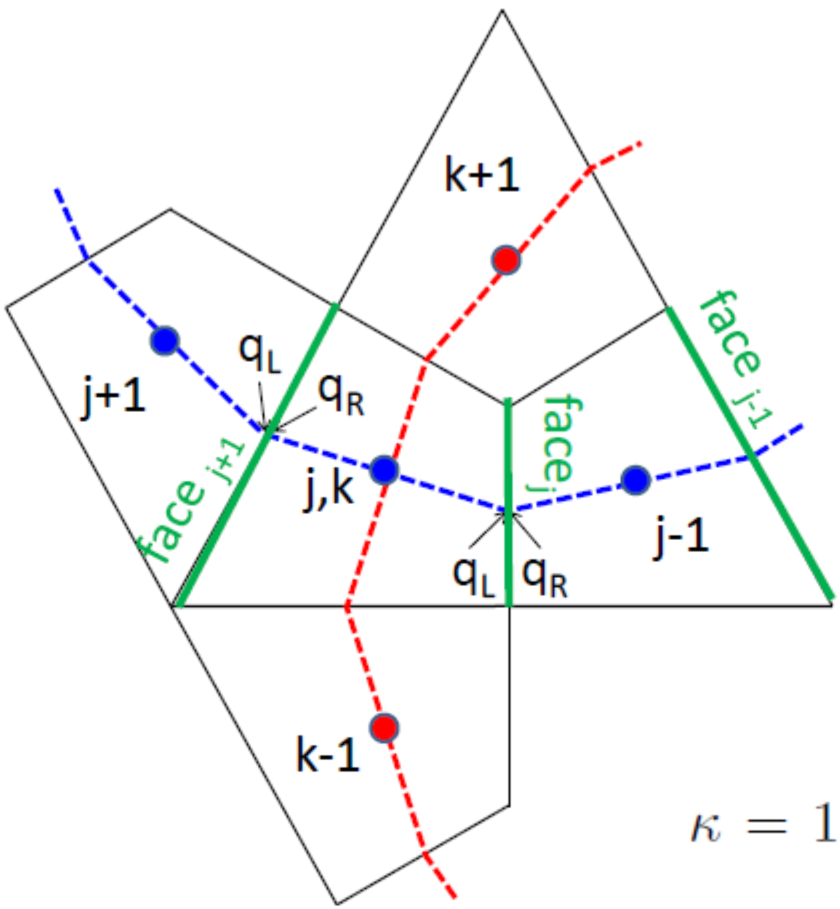
Table 1. Data structures that enable Hamiltonian Path based discretization

face[:,i] = [node1, node2, leftCell , rightCell, leftCellFaceNumber, rightCellFaceNumber]



Can be extended to 3-D quite easily

Spatial Discretization



Popular MUSCL type reconstruction with Koren's differentiable limiter

$$P_L = \left(1 + \frac{\Psi_j}{4}[(1-\kappa)\nabla + (1+\kappa)\Delta]\right) p_j$$

$$P_R = \left(1 - \frac{\Psi_{j-1}}{4}[(1-\kappa)\nabla + (1+\kappa)\Delta]\right) p_{j+1}$$

$$\Psi_j = \frac{3\Delta p_j \nabla p_j + \epsilon}{2(\Delta p_j - \nabla p_j)^2 + 3\Delta p_j \nabla p_j + \epsilon}$$

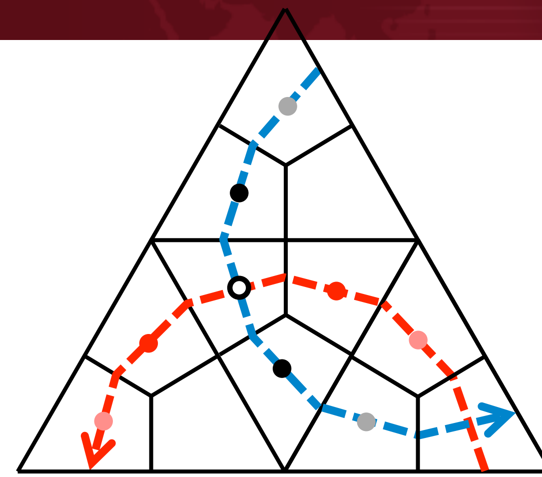
10^{-10}

$\kappa = 1/3$ **gives a 3rd order scheme on regular grids**

*Δ and ∇ are backward and forward difference operators

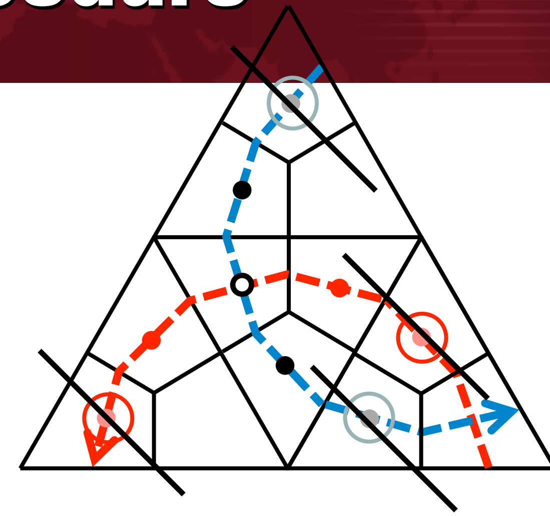
Implicit solution procedure

$$\left[I + \Delta\tau \frac{\partial R_i}{\partial q_j} \right] \Delta q_j = -\Delta\tau R(q_i)$$



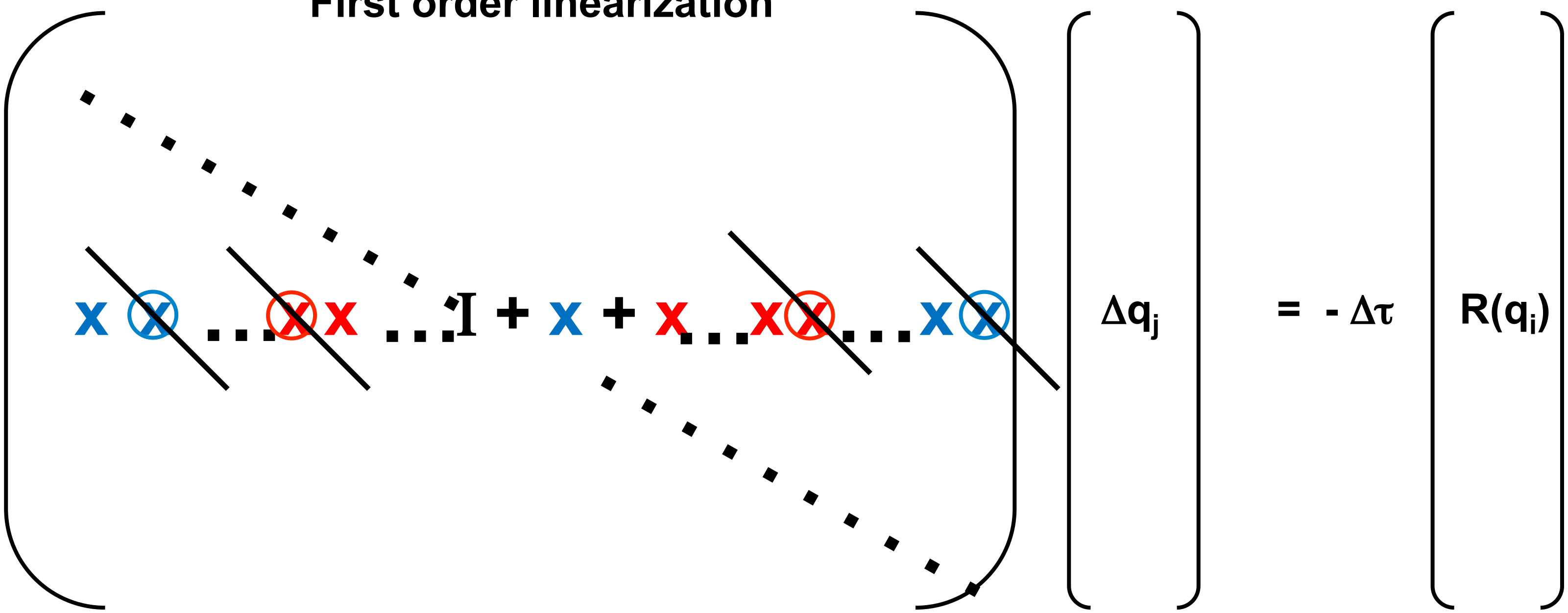
$$\begin{pmatrix} \ddots & & \\ x \;\; x \;\; \ldots x \;\; x \;\; \ldots I + x + x \ldots x \;\; x \ldots x \;\; x \\ & & \ddots \end{pmatrix} \begin{pmatrix} \Delta q_j \end{pmatrix} = -\Delta\tau \begin{pmatrix} R(q_i) \end{pmatrix}$$

Implicit solution procedure



$$\left[I + \Delta\tau \frac{\partial R_i}{\partial q_j}\right] \Delta q_j = -\Delta\tau R(q_i)$$

First order linearization



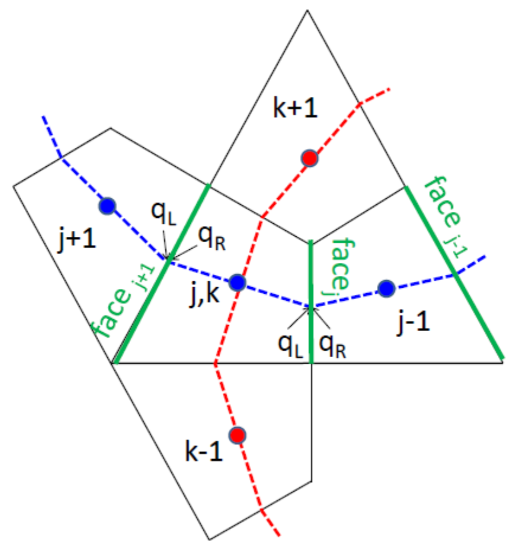
Line Gauss-Seidel

$$\Delta\tau \left(\frac{\partial \hat{F}_j}{\partial q_L} - \frac{\partial \hat{F}_{j+1}}{\partial q_R} \right) \quad \Delta\tau \left(\frac{\partial \hat{F}_k}{\partial q_L} - \frac{\partial \hat{F}_{k+1}}{\partial q_R} \right)$$

$$\left[I + x + x \;\vdots\; x \;\vdots\; x \;\vdots\; x \;\vdots\; x \right] \begin{bmatrix} \Delta q \\ \Delta q^{+} \\ \Delta q^{-} \\ \Delta q^{+} \\ \Delta q^{-} \end{bmatrix} = -\Delta\tau \; R(q)$$

$$\Delta\tau \left(-\frac{\partial \hat{F}_{k+1}}{\partial q_L} \quad \frac{\partial \hat{F}_k}{\partial q_R} \quad -\frac{\partial \hat{F}_{j+1}}{\partial q_L} \quad \frac{\partial \hat{F}_j}{\partial q_R} \right)$$

First order linearization



$$\left[I + x + x \;\vdots\; x \;\vdots\; x \right] \begin{bmatrix} \Delta q \\ \Delta q^{+} \\ \Delta q^{-} \end{bmatrix} = -\Delta\tau \, R(q) - \left[x \;\vdots\; x \right] \begin{bmatrix} \Delta q^{+} \\ \Delta q^{-} \end{bmatrix}$$

Diagonally Dominant Line Gauss Seidel (Buelow (2001))

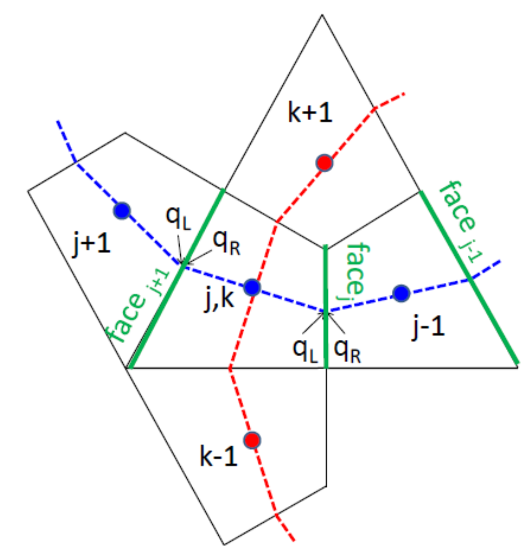
Approximate Factorization (ADI)

$$\Delta\tau\left(\boxed{\frac{\partial \hat{F}_j}{\partial q_L} - \frac{\partial \hat{F}_{j+1}}{\partial q_R}}\right) \qquad \Delta\tau\left(\boxed{\frac{\partial \hat{F}_k}{\partial q_L} - \frac{\partial \hat{F}_{k+1}}{\partial q_R}}\right)$$

$$\left[\begin{array}{c|c|c|c|c} I + x + x & x & x & x & x \end{array}\right] \begin{pmatrix} \Delta q \\ \Delta q^+ \\ \Delta q^- \\ \Delta q^+ \\ \Delta q^- \end{pmatrix} = -\Delta\tau \;\; R(q)$$

$$\Delta\tau\left(-\frac{\partial \hat{F}_{k+1}}{\partial q_L} \quad \frac{\partial \hat{F}_k}{\partial q_R} \quad -\frac{\partial \hat{F}_{j+1}}{\partial q_L} \quad \frac{\partial \hat{F}_j}{\partial q_R}\right)$$

First order linearization



Can be written as:

$$\left[\begin{array}{c|c|c|c|c} I + x & x & x & 0 & 0 \end{array}\right] \underbrace{\begin{pmatrix} I + x & 0 & 0 & x & x \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta q \\ \Delta q^+ \\ \Delta q^- \\ \Delta q^+ \\ \Delta q^- \end{pmatrix}}_{\Delta q'} = -\Delta\tau \;\; R(q)$$

Factorization is approximate as higher order terms have to be neglected

Alternating Direction Implicit
(Peacement-Rachford, Douglas)

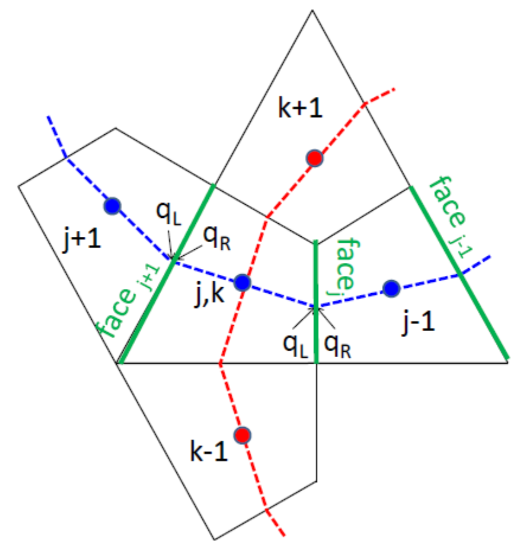
Approximate Factorization (DDADI)

$$\Delta\tau\left(\frac{\partial \hat{F}_j}{\partial q_L} - \frac{\partial \hat{F}_{j+1}}{\partial q_R}\right) \quad \Delta\tau\left(\frac{\partial \hat{F}_k}{\partial q_L} - \frac{\partial \hat{F}_{k+1}}{\partial q_R}\right)$$

$$\begin{bmatrix} I + x + x & x & x & x & x \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta q^+ \\ \Delta q^- \\ \Delta q^+ \\ \Delta q^- \end{bmatrix} = -\Delta\tau \; R(q)$$

$$\Delta\tau\left(-\frac{\partial \hat{F}_{k+1}}{\partial q_L} \quad \frac{\partial \hat{F}_k}{\partial q_R} \quad -\frac{\partial \hat{F}_{j+1}}{\partial q_L} \quad \frac{\partial \hat{F}_j}{\partial q_R} \right)$$

First order linearization



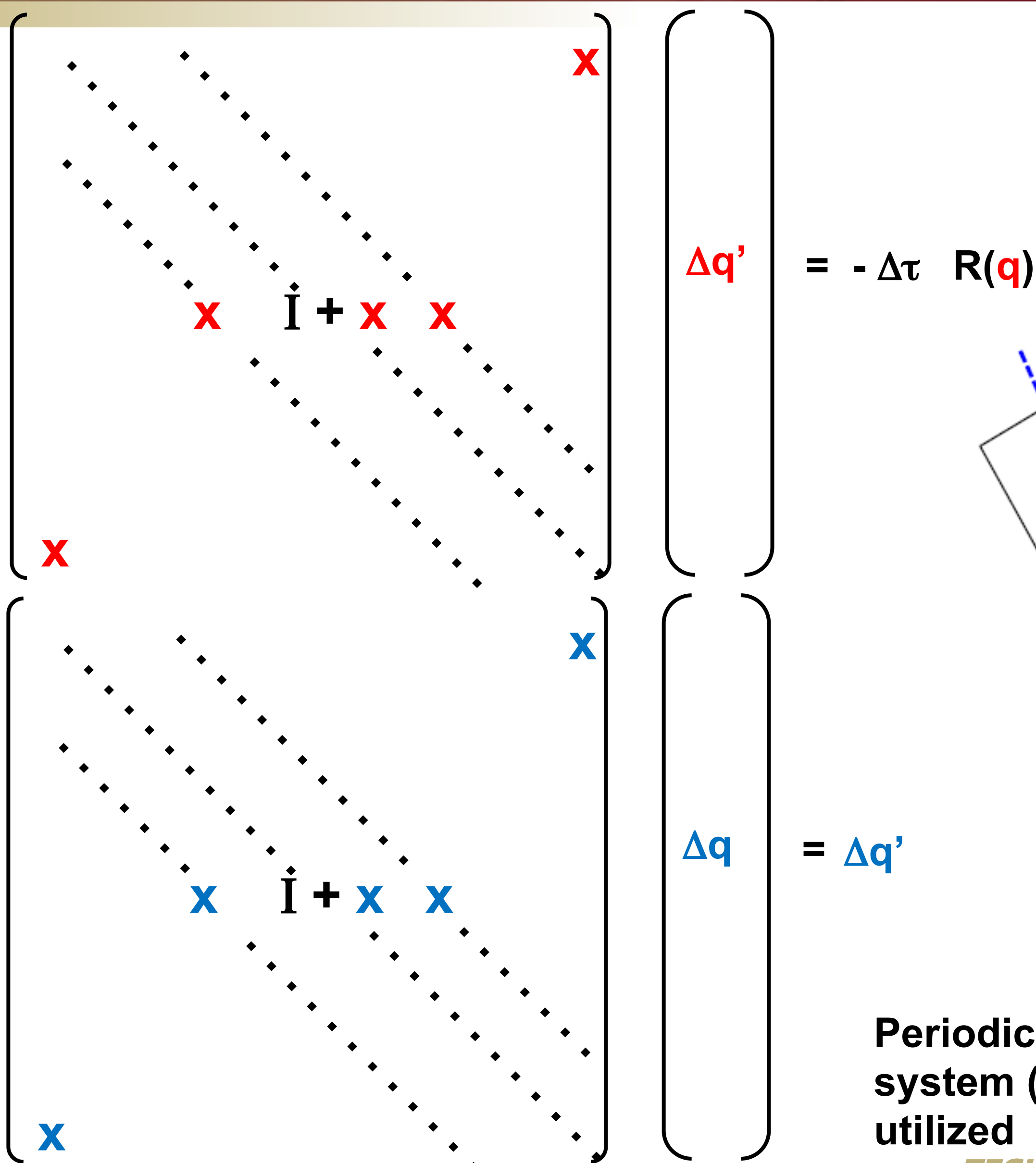
Can be written as:

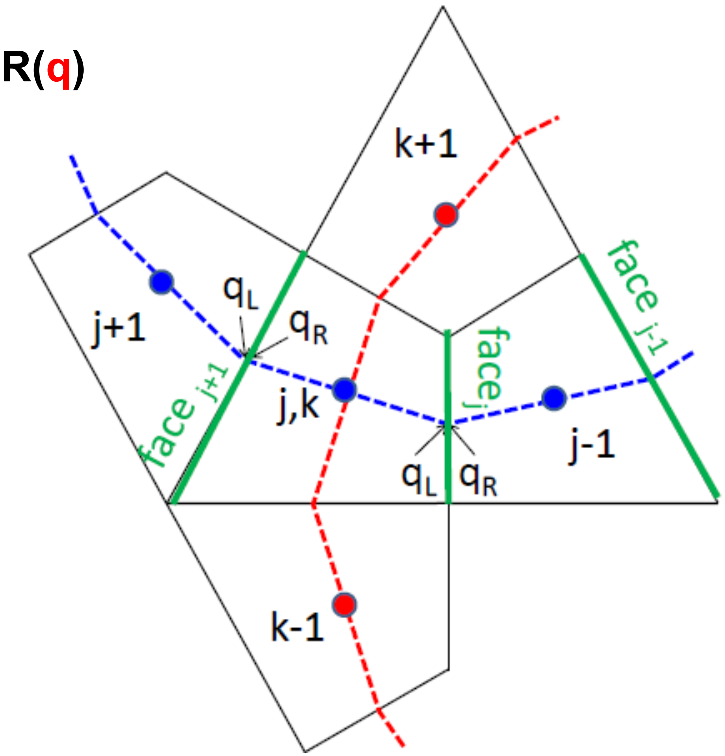
$$\begin{bmatrix} I + x + x & x & x & 0 & 0 \end{bmatrix} \quad (I + x + x)^{-1} \underbrace{\begin{bmatrix} I + x + x & 0 & 0 & x & x \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta q^+ \\ \Delta q^- \\ \Delta q^+ \\ \Delta q^- \end{bmatrix}}_{\Delta q'} = -\Delta\tau \, R(q)$$

Factorization is approximate as higher order terms has to be neglected

<u>D</u>iagonally <u>D</u>ominant <u>A</u>lternating <u>D</u>irection <u>I</u>mplicit (Pulliam)

Inversion of banded matrices



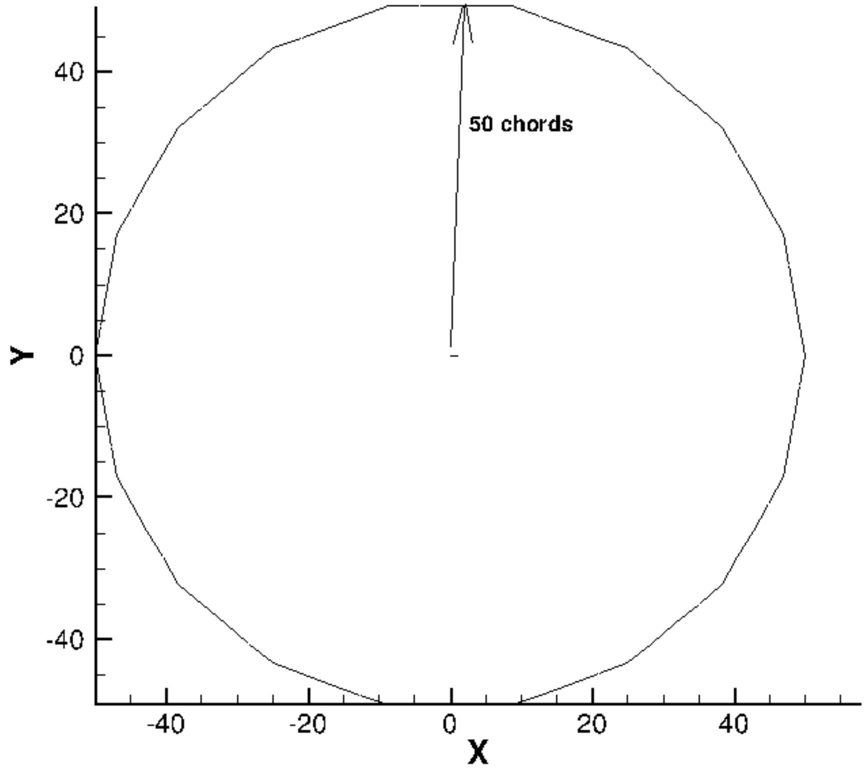


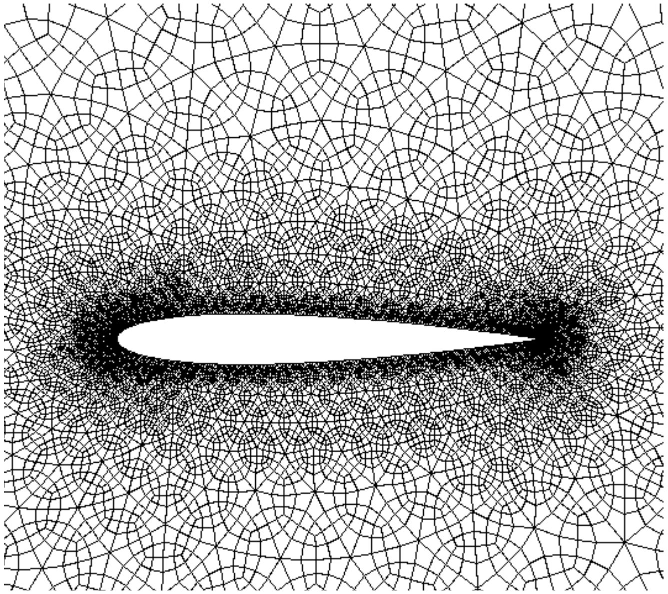
Periodic version of block-tridiagonal system (modified Thomas algorithm) is utilized

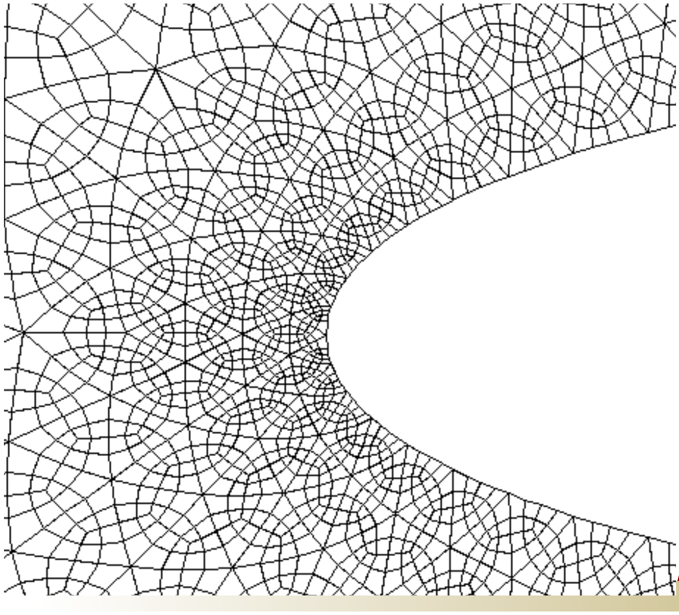
Solver Architecture

- Residual calculation
  - For each Chain:
    - Collect all faces forming loops (preprocessed data structure)
    - Collect all the cells forming loops
    - Reconstruct left and right states using favorite reconstruction scheme (MUSCL in this case)
    - Use Riemann solver (Roe in this case) to find face fluxes
    - Compute viscous fluxes by finite differencing
    - Add face fluxes at each face to corresponding cells
- Inversion
  - For each Color:
    - For Chains in each color
      - Collect all faces forming loops
      - Find Left and Right state Jacobians
      - Add/subtract contributions to cells to create a banded block system (periodic) for closed chains
      - Invert block tridiagonal (considering 1st order LHS) system using Thomas algorithm or periodic variant for each chain
      - Update right hand side with result from the inversion

Results : Transonic Airfoil





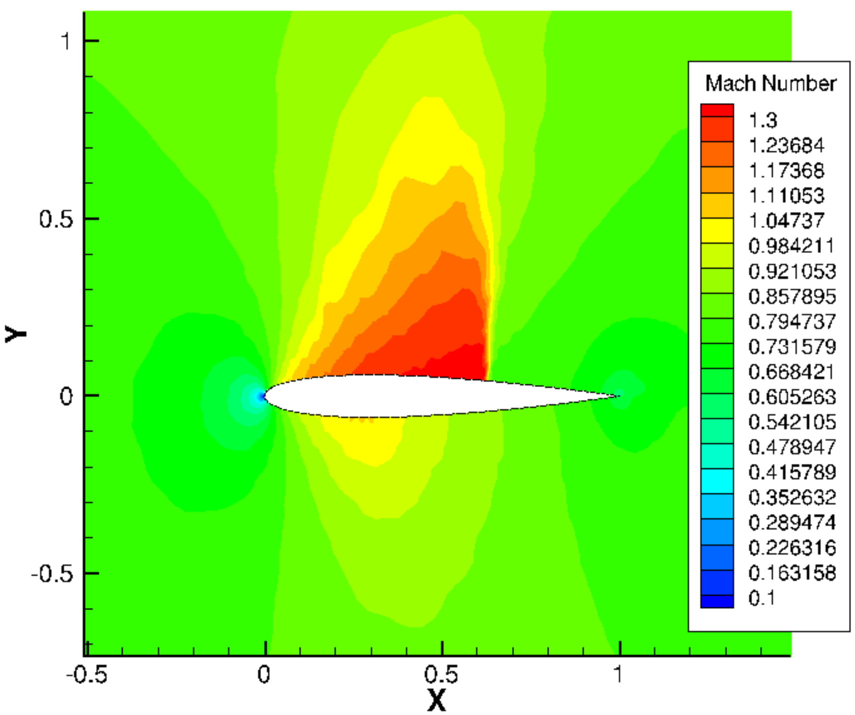


1732 triangles in the original mesh

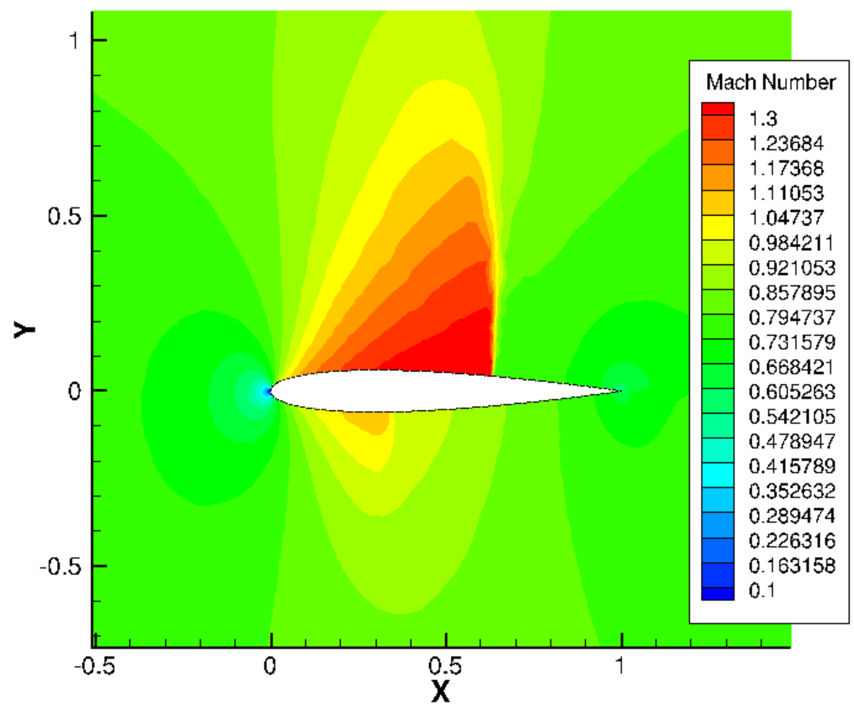
20748 quadrilaterals created after sub-division

Analytical NACA0012 profile used for surface point insertion

Results: Transonic Airfoil



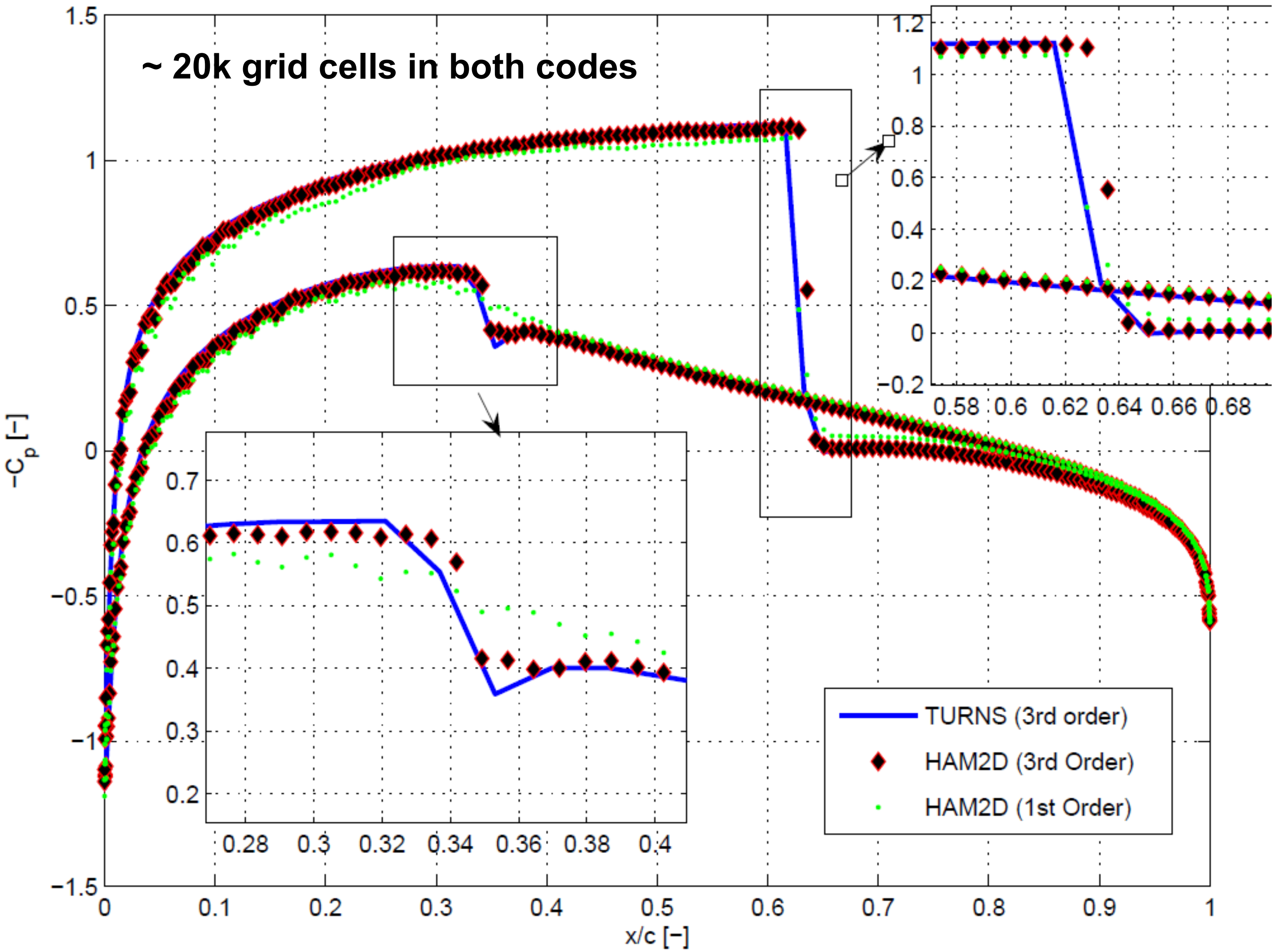
1st order solution



Solution with MUSCL reconstruction

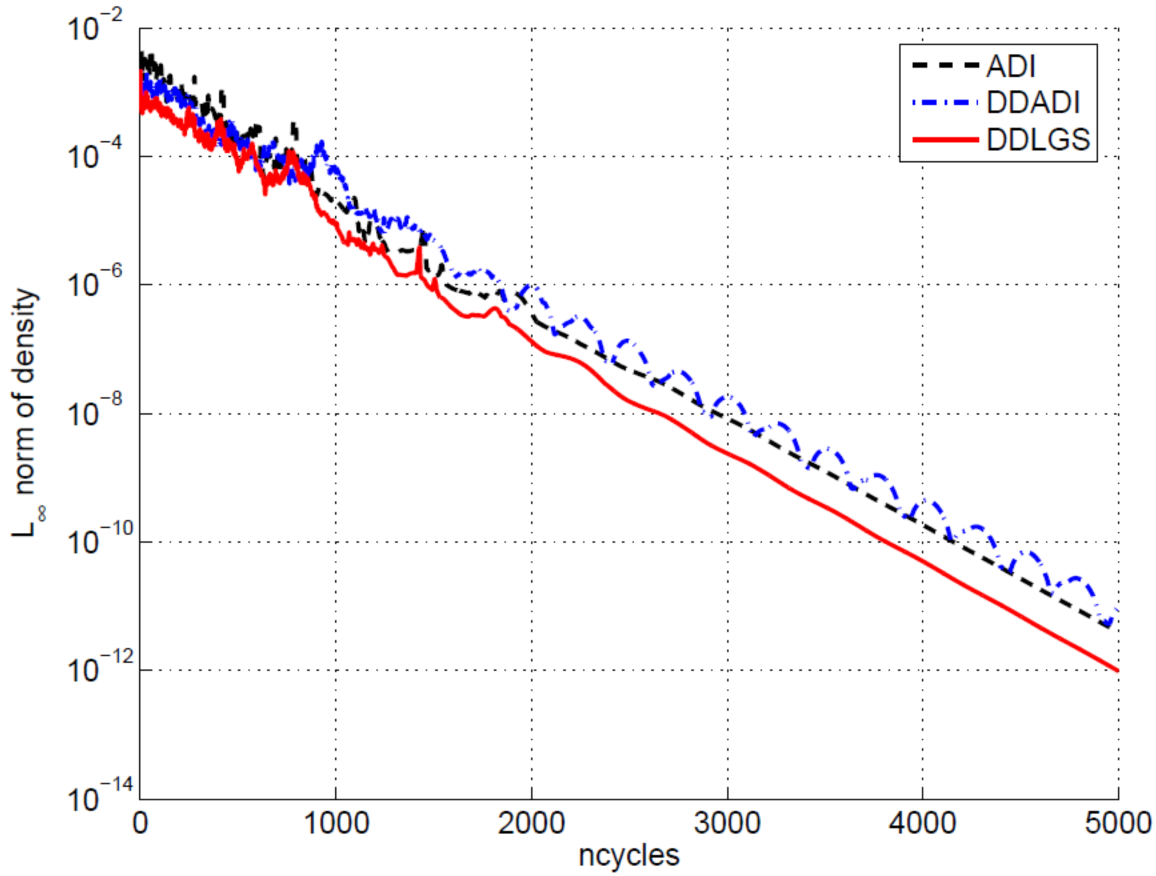
MUSCL takes 3% longer on per iteration basis and requires 10 % more non-linear cycles for same level of convergence

Results: Transonic Airfoil



TURNS – structured grid based code originally developed by NASA/U.S Army and widely used at many Universities

Results: Transonic Airfoil (Convergence of Linf norm)



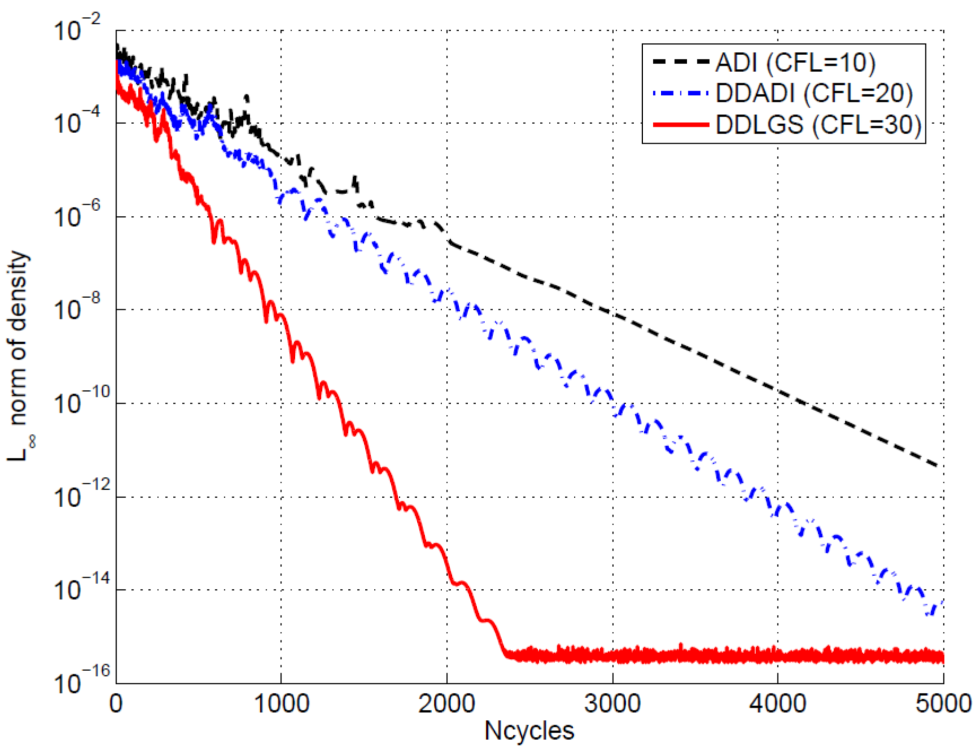
(a) Residual vs ncycles for CFL=10

No linear solver and linear sweeps

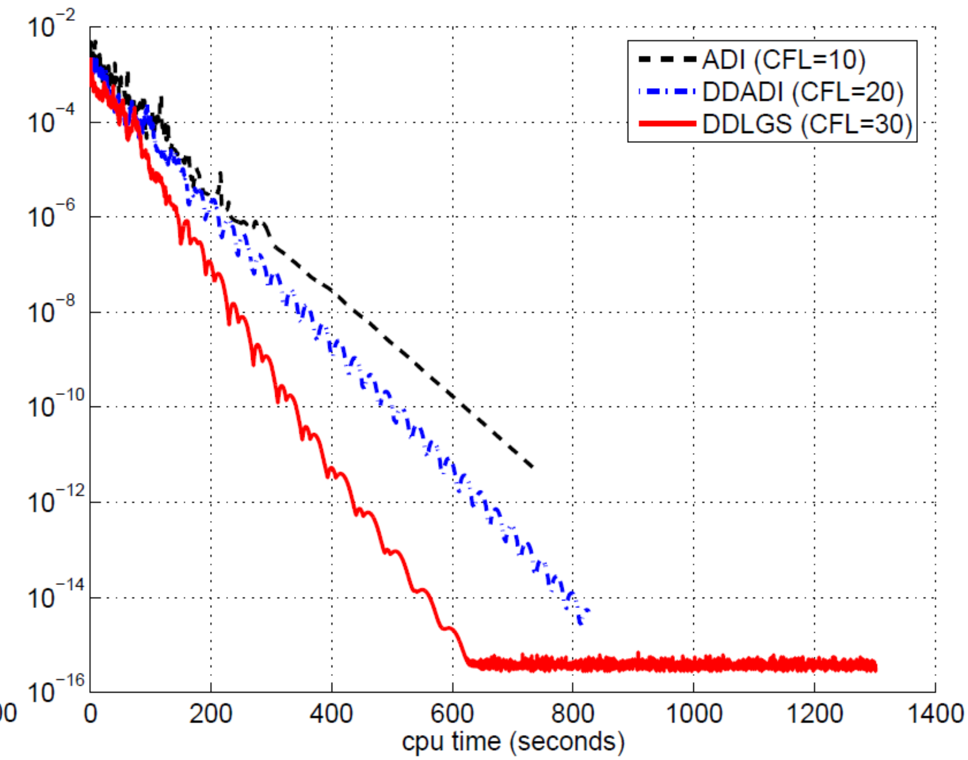
Each cycle corresponds to exactly one non-linear iteration and one residual evaluation

No CFL ramping or tuning, CFL fixed from the beginning

Results: Transonic Airfoil (Convergence of Linf norm)



(b) Residual vs ncycles for max CFL

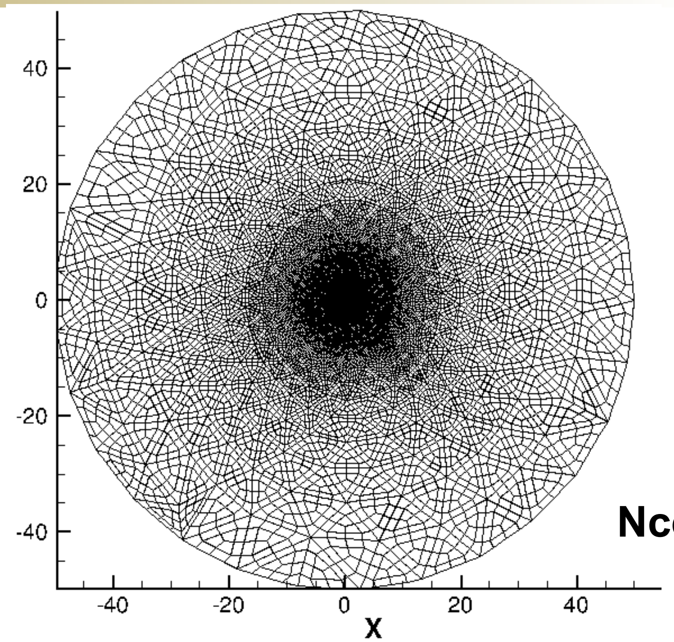


(c) Residual vs cpu time for max CFL

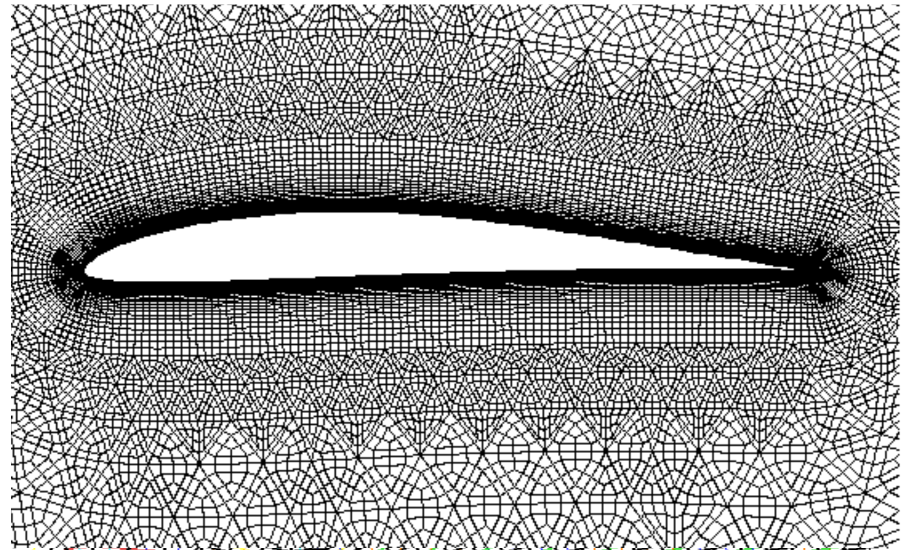
Similar convergence trends obtained by Buelow et al

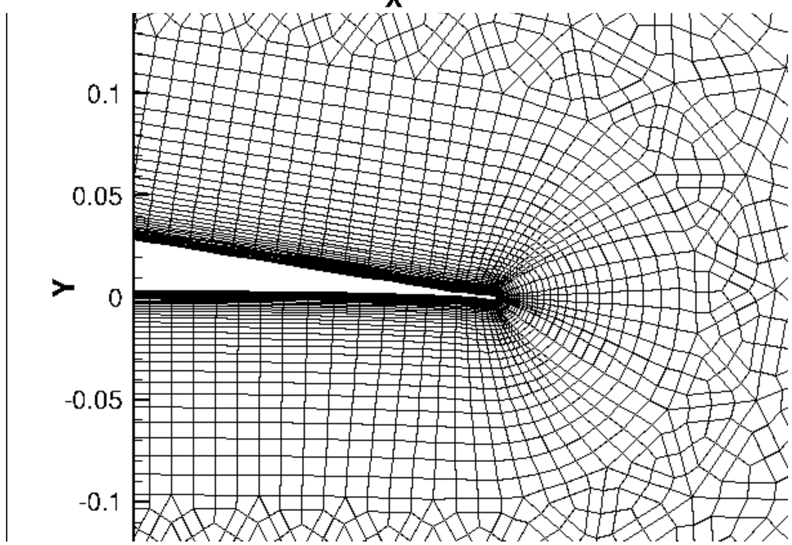
(computer and fluids, 2001 on structured grids)

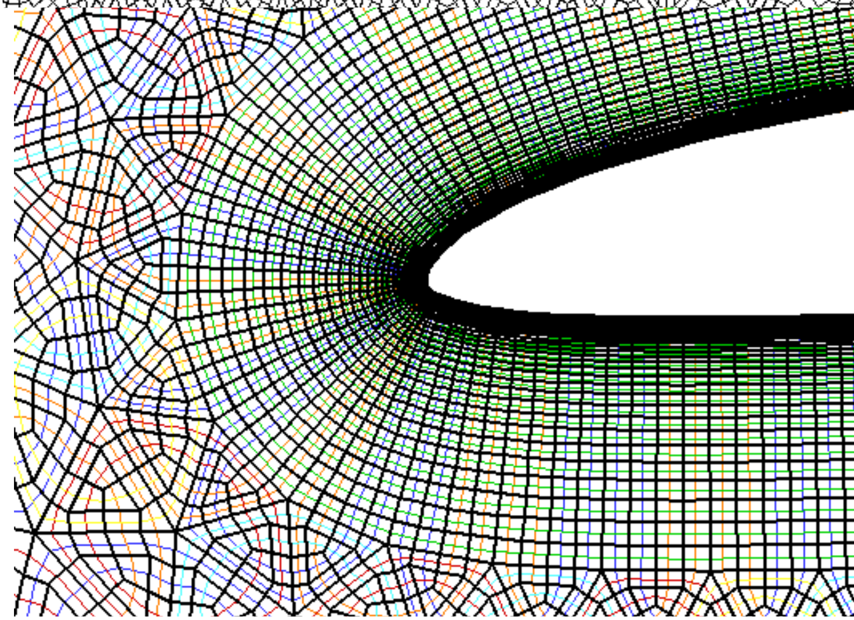
Viscous Airfoil
(Eppler 387, Re=60,000, M=0.1)



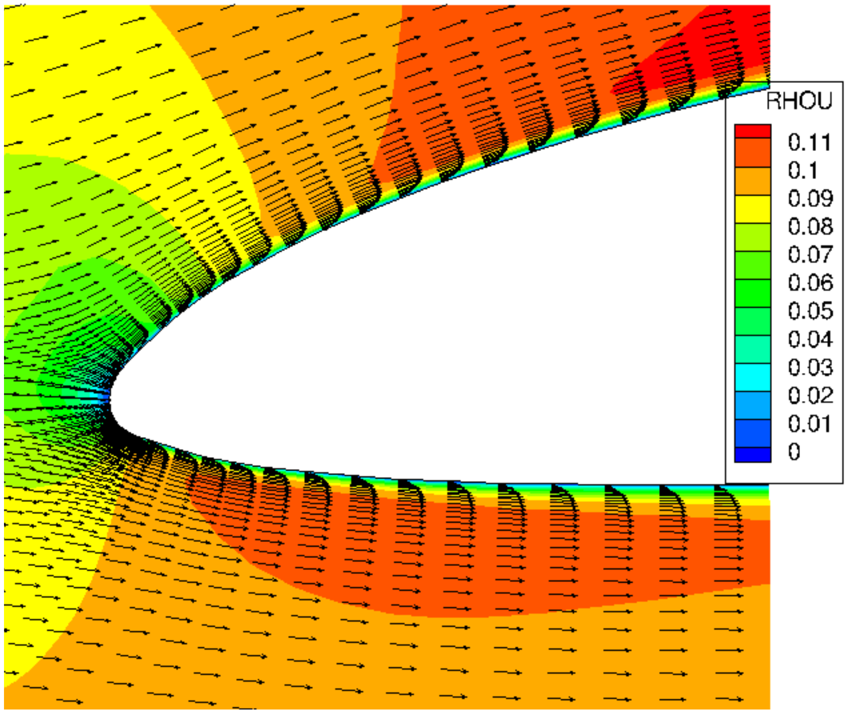
Ncells ~ 50k



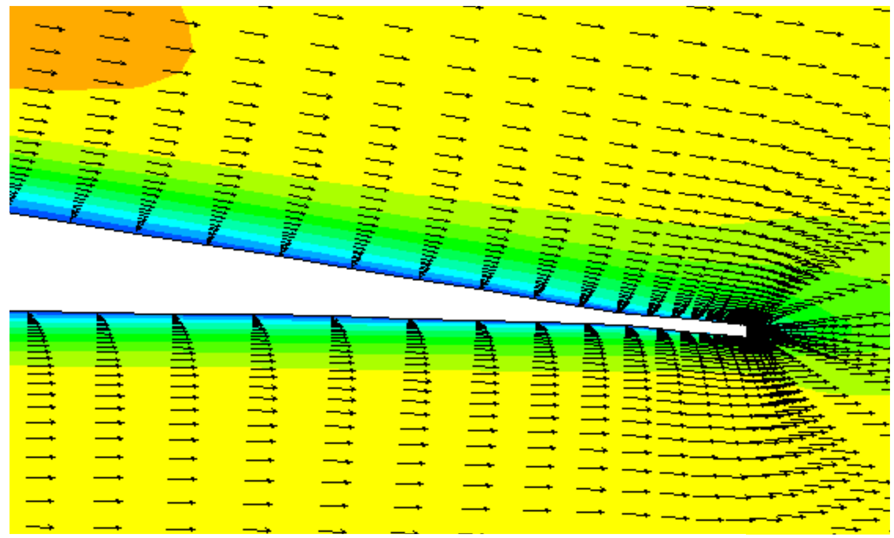


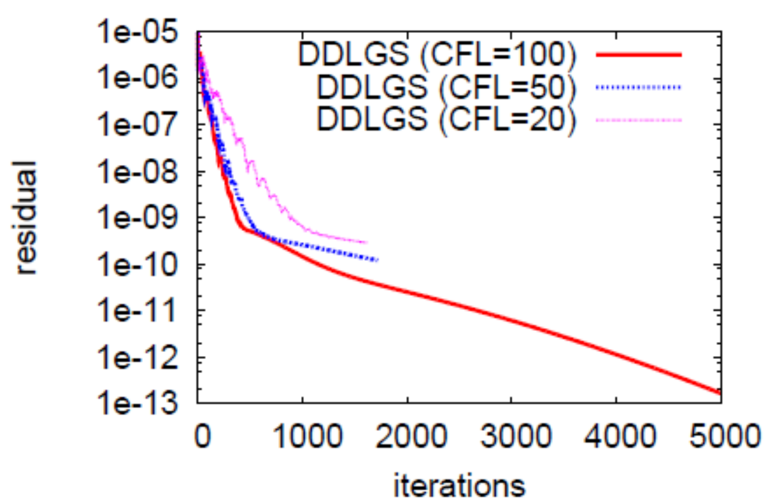


Viscous Airfoil (Eppler 387, Re=60,000, M=0.1)



Ncells ~ 50k





Physically reasonable solutions, comprehensive validation needed

Convergence slows after the first four orders

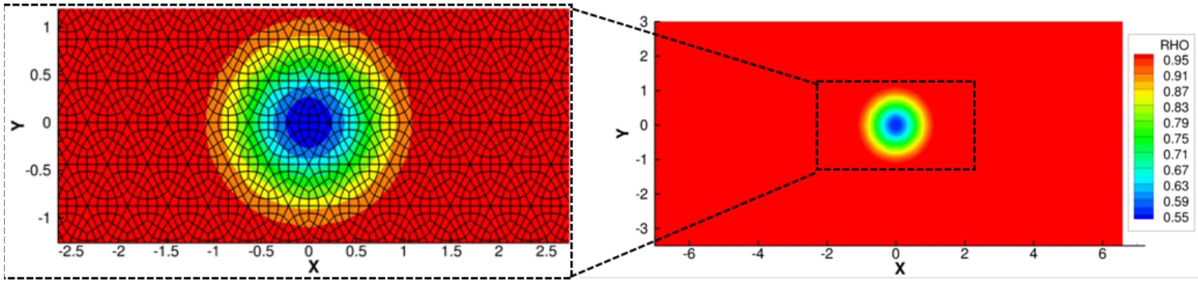
Possibly due to lower frequency errors that have smaller damping

Questions on viability of the approach

- How would reconstruction schemes such as WENO5 and compact WENO fare on Hamiltonian grids?
- Is the solution quality and convergence strongly dependent on the curvature of Hamiltonian paths? Can we control the curvature?
- Can this be extended to make a functional 3-D RANS solver? With Hamiltonian loops on the surface and strands in the wall normal direction.. Will the results be accurate?
- Can this approach be parallelized? Will resulting code be scalable?
- Can the method be used in an overset framework such as HELIOS?
- Can the Hamiltonian loop approach be extended to general surface tesselations?
- How will new points inside cells be introduced such that they flush with the surface (same question as for high-order FE methods)?
- …

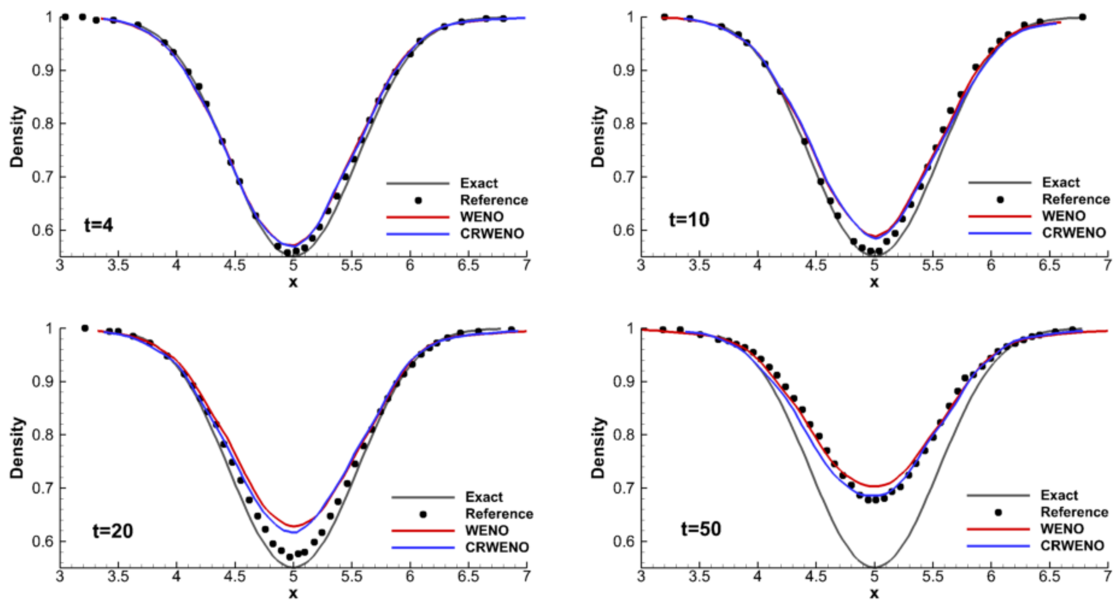
Jim Baeder's group at Univ of MD in collaboration with Helios development team has answered most of these questions affirmatively

Isentropic Vortex Convection with WENO schemes



Grid and initial density contours

- 11,700 quadrilateral cells.
- with 10 subiterations
- Temporal method : 2^{nd} order BDF
- Spatial method : 5^{th} order



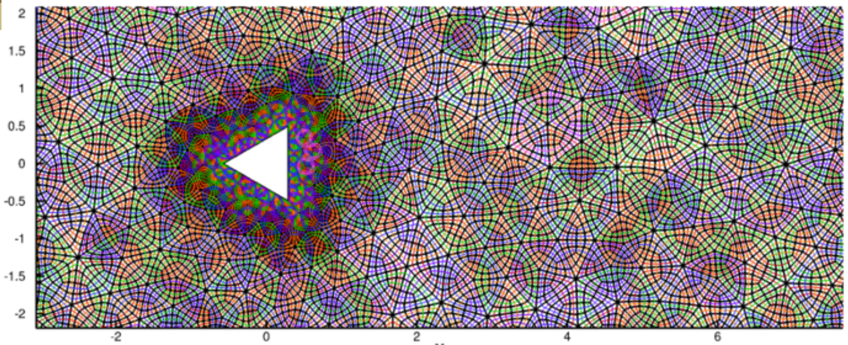
Density profiles across the vortex core at different solution times

Comparison results with DG method.

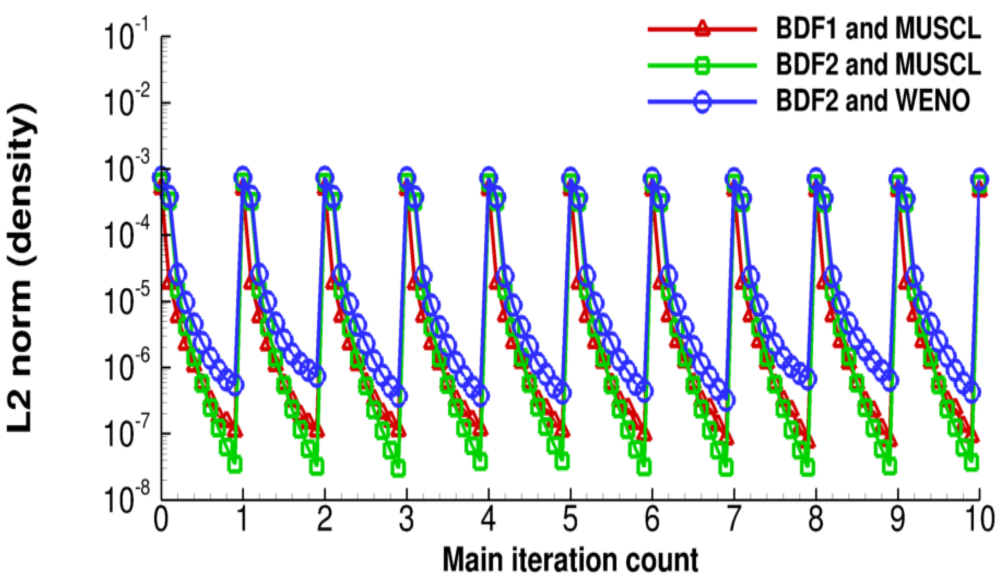
- More dissipated vortex core until
- Similar density profile at
- 5^{th} order CRWENO shows better conserved core strength than 5^{th} order WENO.

Jung et al, AIAA SciTech 2016

Vortex Shedding over Wedge

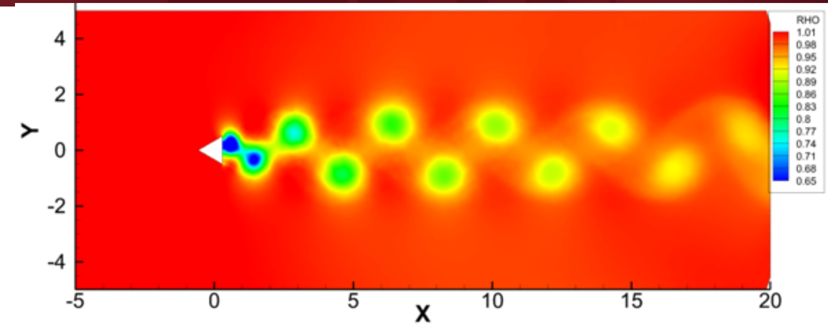


Mesh system around the wedge

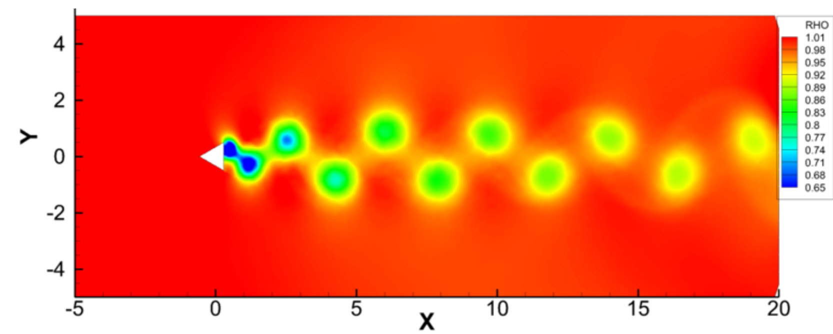


Unsteady residual convergence during sub-iteration

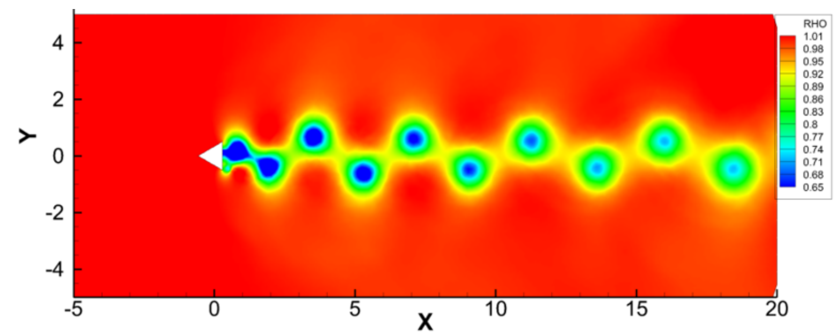
Jung et al. AIAA SciTech 2016



BDF1 and MUSCL



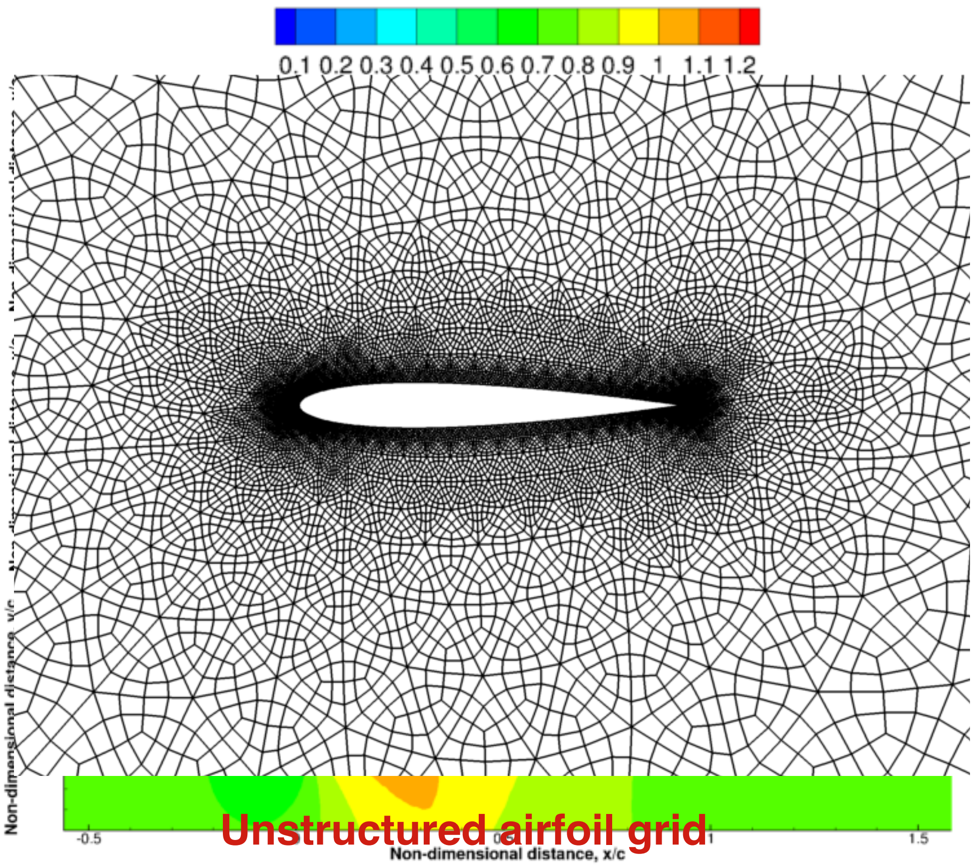
BDF2 and MUSCL



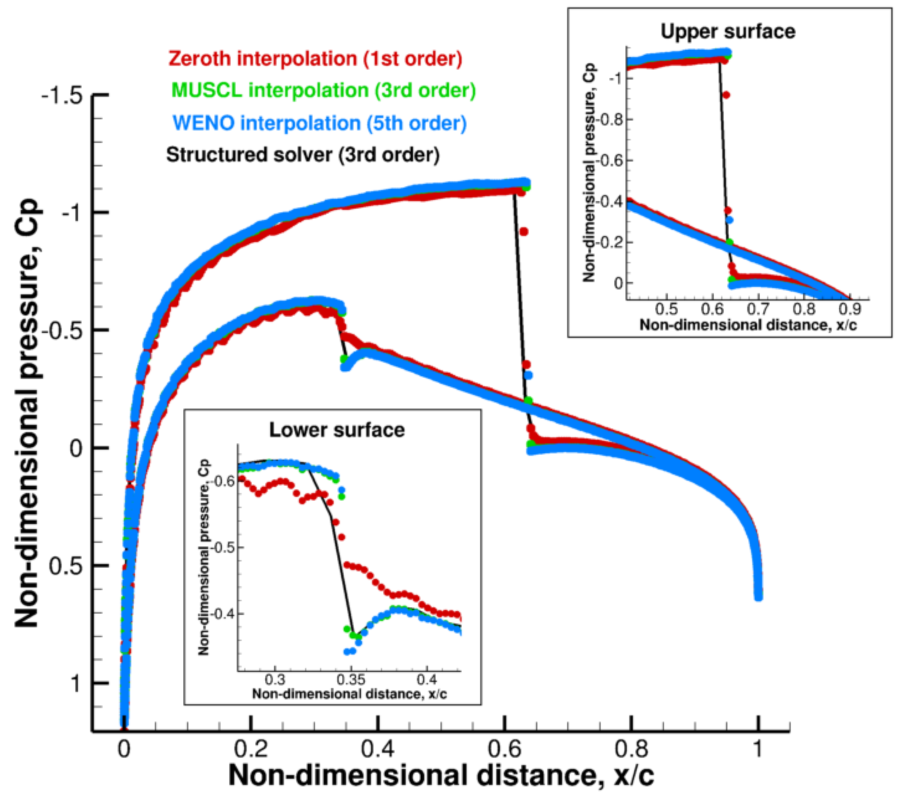
BDF2 and WENO

Transonic Flow Over Airfoil WENO comparison to MUSCL

- Inviscid NACA 0012 at $M = 0.8$, AoA = 1.25^0
- 1,732 triangles with a total of 20,784 quad cells
- Pressure distribution compared to a structured solver (TURNS-2D)



Unstructured airfoil grid



Surface pressure distribution

Govindarajan et al. AHS 2015

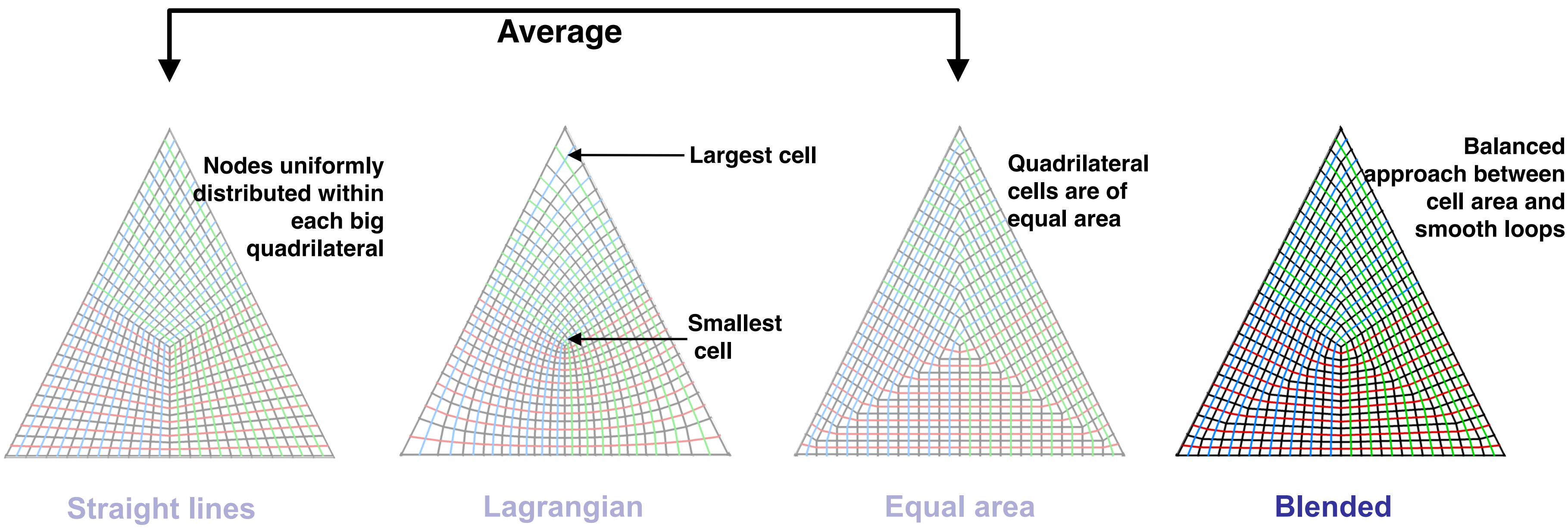
UNCLASSIFIED

Questions on viability of the approach

- How would reconstruction schemes such as WENO5 and compact WENO fare on Hamiltonian grids?
- **Is the solution quality and convergence strongly dependent on the curvature of Hamiltonian paths? Can we control the curvature?**
- Can this be extended to make a functional 3-D RANS solver? With Hamiltonian loops on the surface and strands in the wall normal direction.. Will the results be accurate?
- Can this approach be parallelized? Will resulting code be scalable?
- Can the method be used in an overset framework such as HELIOS?
- Can the Hamiltonian loop approach be extended to general surface tesselations?
- How will new points inside cells be introduced such that they flush with the surface (same question as for high-order FE methods)?
- …

Mesh Smoothing

- **Smoothing of cells and loops improves accuracy**
- Interior nodes can be positioned in a few different ways:

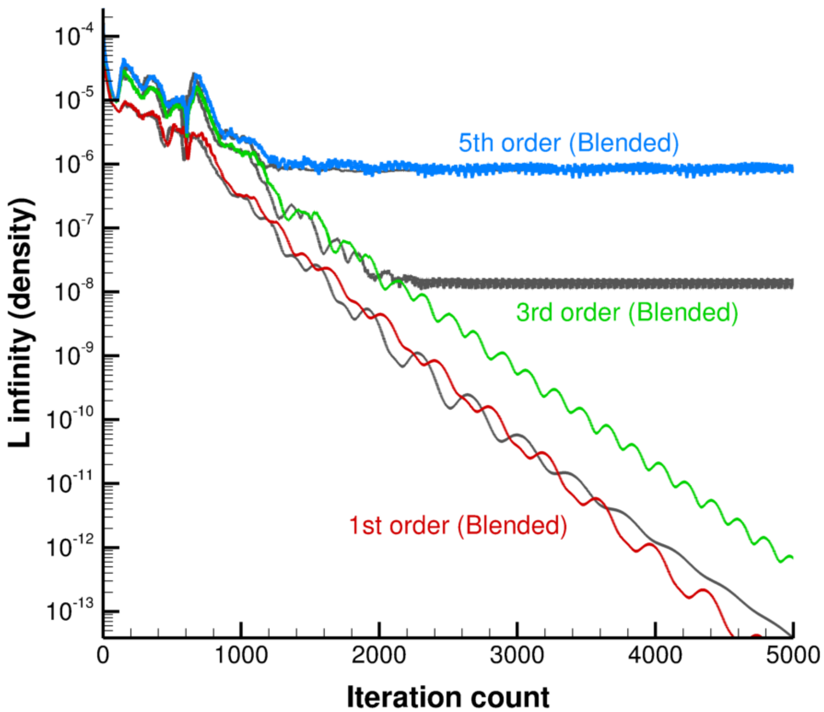


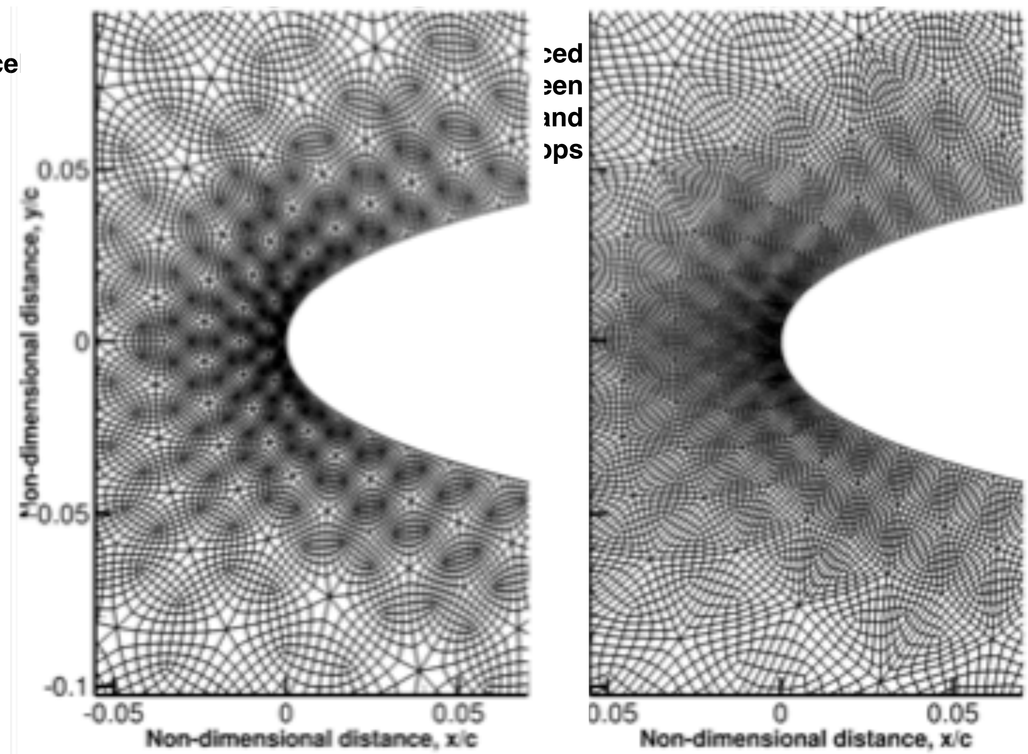
- The appropriated smoothing techniques are mentioned in subsequent results
- **Most successful were the Lagrangian and blended techniques**

Govindarajan et al. AHS 2015

Transonic Flow Over Airfoil: Mesh

- Residual convergence between mesh smoothing techniques (Lagrangian and blended)
- Flow conditions as shown before (M = 0.8, AoA = 1.25^0)
- Blended mesh aids in the convergence of third-order MUSCL
- Fifth-order shows a stalled behaviour – oscillation of shock between mesh points





Lagrangian Blended

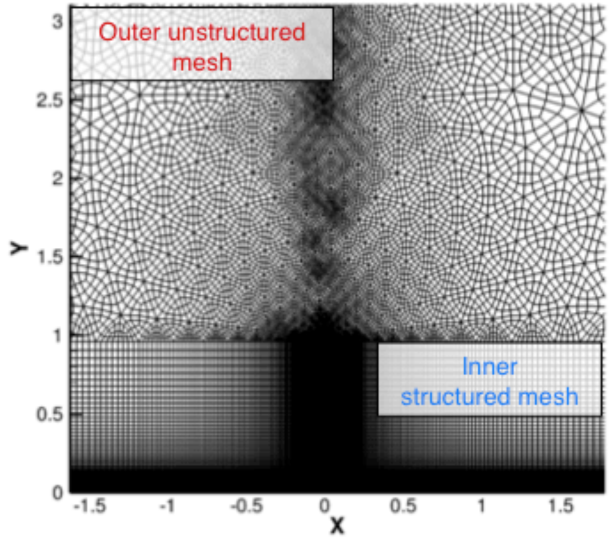
Govindarajan et al. AHS 2015

UNCLASSIFIED

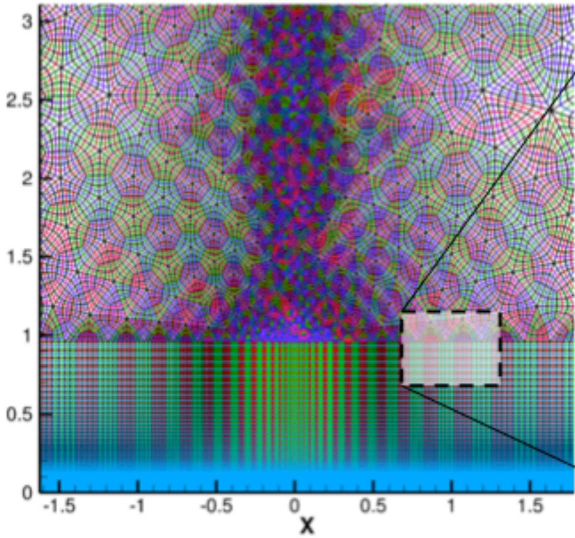
Questions on viability of the approach

- How would reconstruction schemes such as WENO5 and compact WENO fare on Hamiltonian grids?
- Is the solution quality and convergence strongly dependent on the curvature of Hamiltonian paths? Can we control the curvature?
- **Can this be extended to make a functional 3-D RANS solver? With Hamiltonian loops on the surface and strands in the wall normal direction.. Will the results be accurate?**
- Can this approach be parallelized? Will resulting code be scalable?
- Can the method be used in an overset framework such as HELIOS?
- Can the Hamiltonian loop approach be extended to general surface tesselations?
- How will new points inside cells be introduced such that they flush with the surface (same question as for high-order FE methods)?
- …

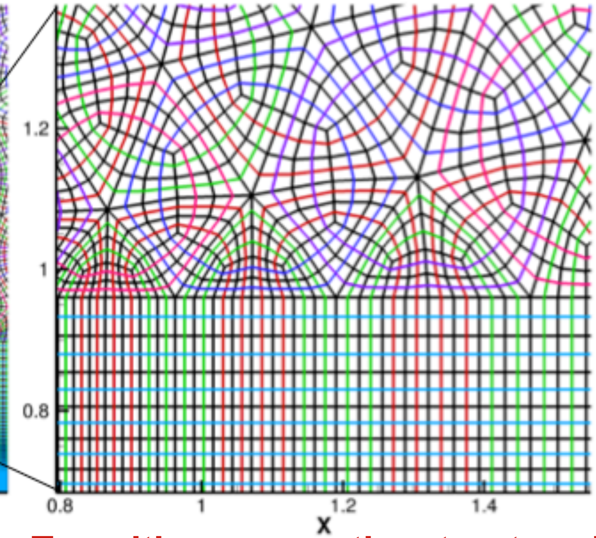
Extension to RANS: Turbulent Flow over Flat Plate



Hybrid mesh system

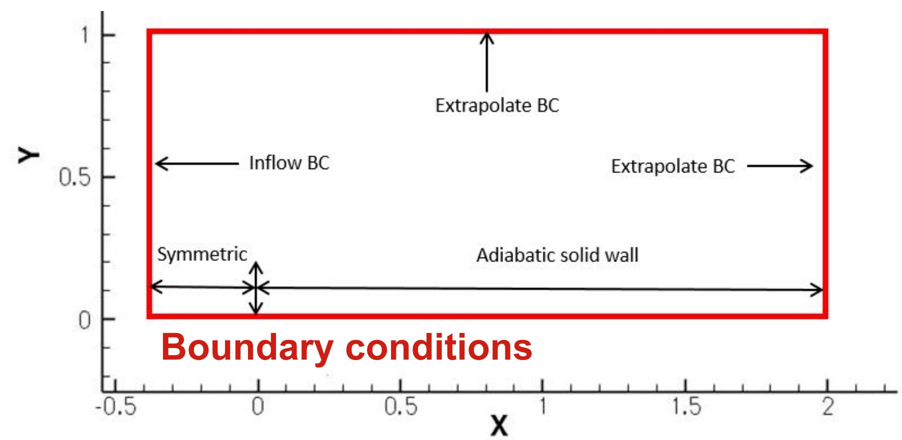


Coloured Hamiltonian paths



Transition across the structured and unstructured mesh

- Hybrid mesh system used to simulate turbulent flow over flat plate
- Calculation of wall distance is trivial
- Hamiltonian loops formed through the structured and unstructured mesh

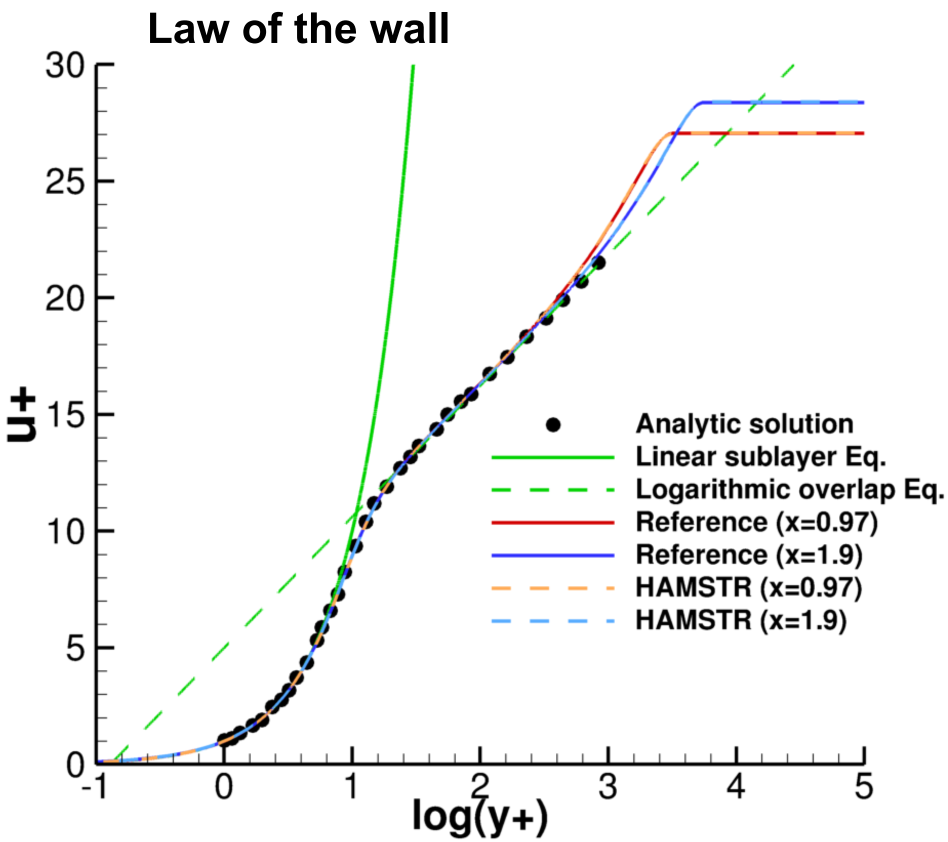


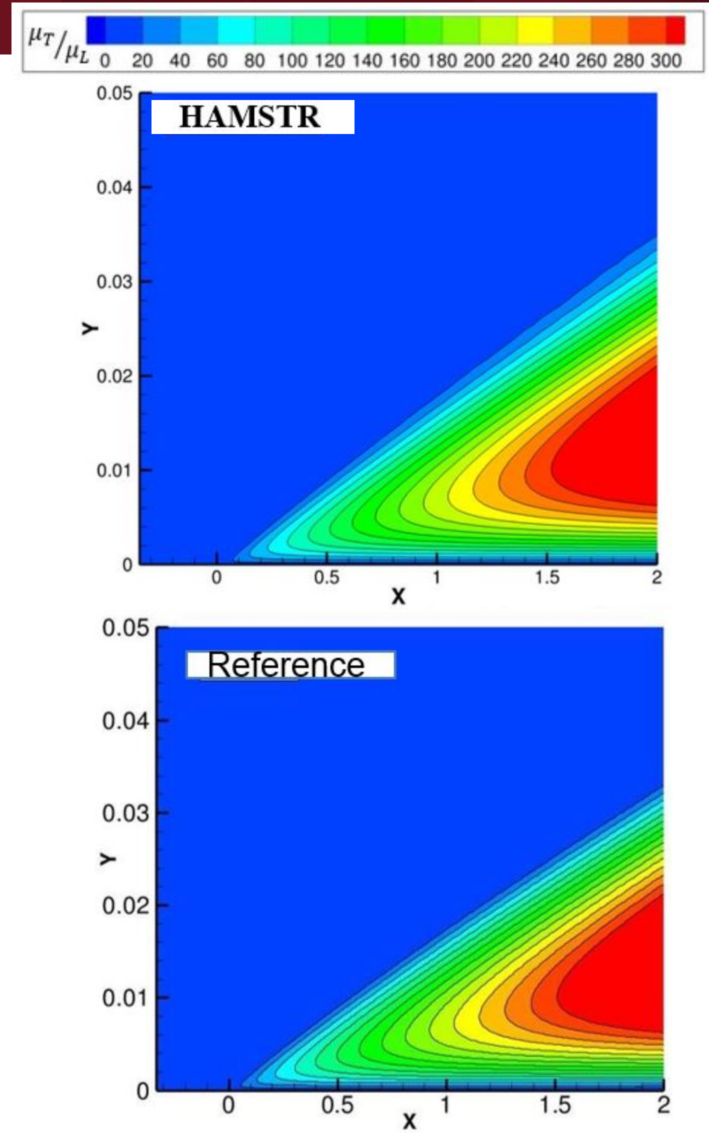
Boundary conditions

Jung et al, AIAA SciTech 2016

UNCLASSIFIED

Turbulent Flow: Flat Plate Profile

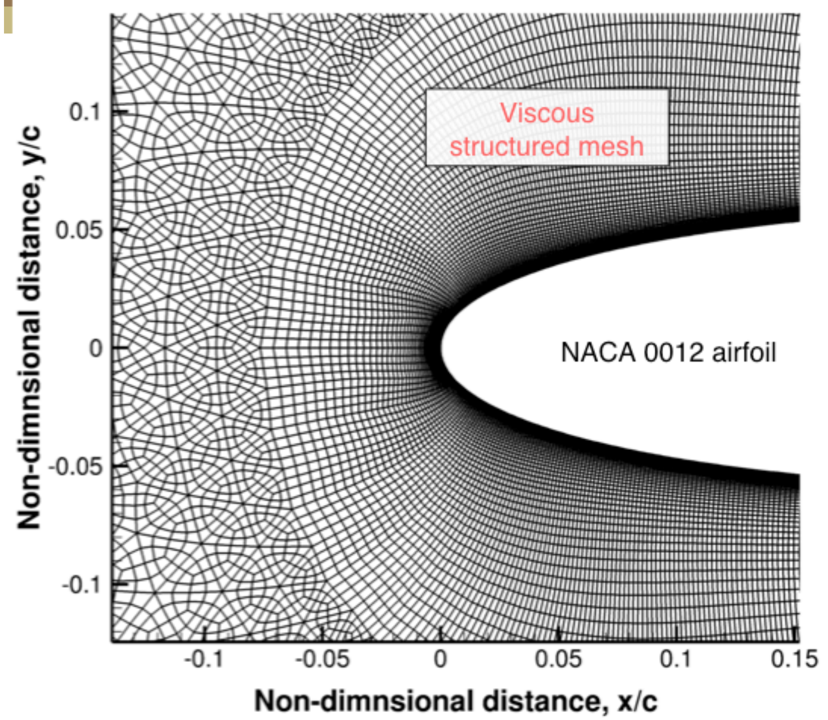


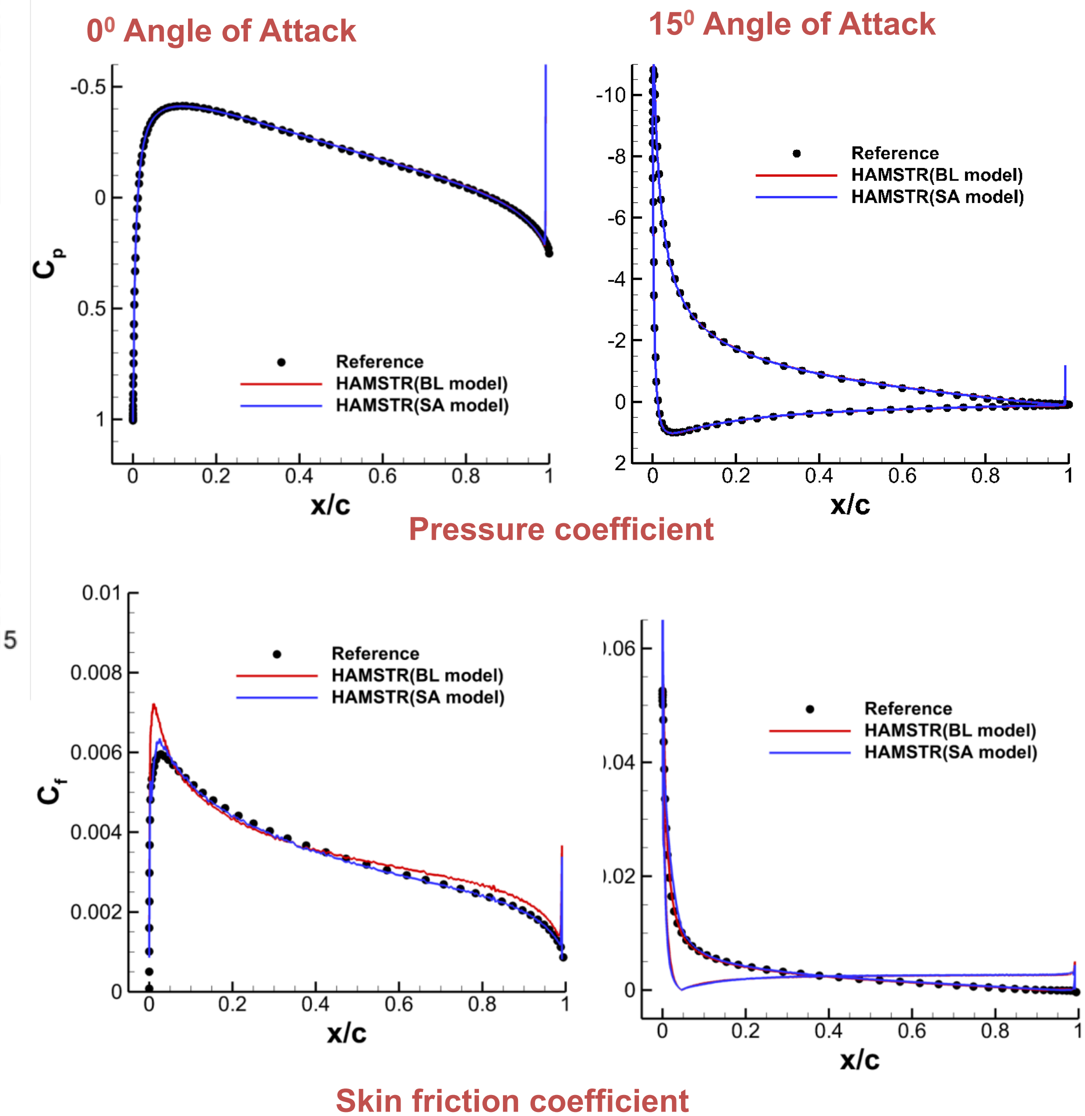


- **Initial spacing , cell count: 117,336**
- **Mach = 0.2, Re = 5,000,000**
- **Good agreement using SA model for both the Cartesian and hybrid HAMSTR mesh**

Jung et al., AIAA SciTech 2016

Turbulent Flow: NACA 0012



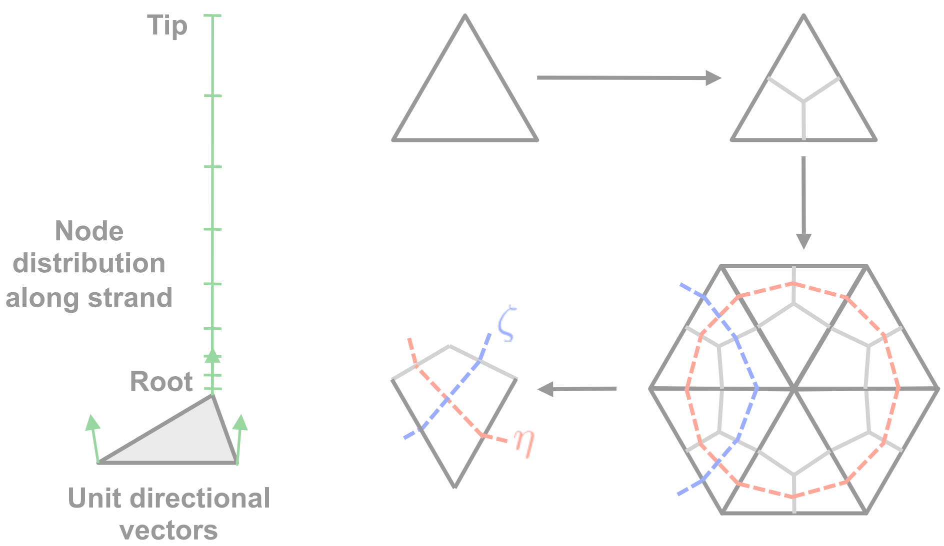


- Initial spacing
- Cell count 112,512
- Mach 0.15, Re 6,000,000

- Good agreement between reference (NASA turbulence site) and HAMSTR

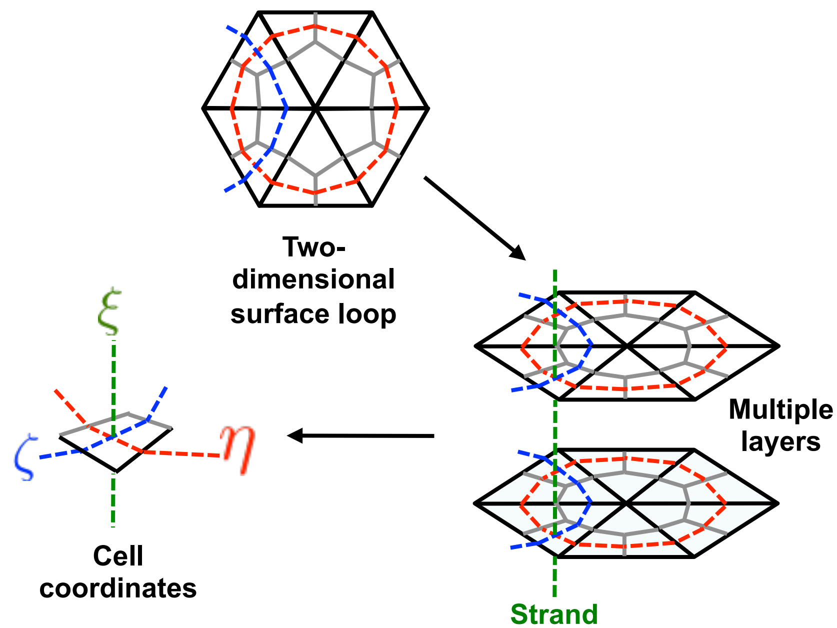
Extension to 3-D

- **Strands grids are employed to extend the formulation to three-dimensions**
- **Formed by extruding the surface mesh in wall normal direction**
- Volume domain formed by "stacking" multiple Hamiltonian path layers
- **Layers are connected with strands and forms the third spatial cell coordinate**



Strand template

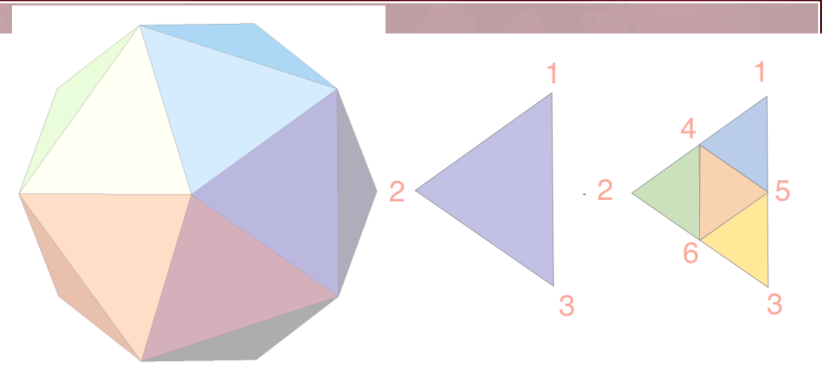
Subdivision of triangle into quadrilateral



Strands and associated cell coordinates

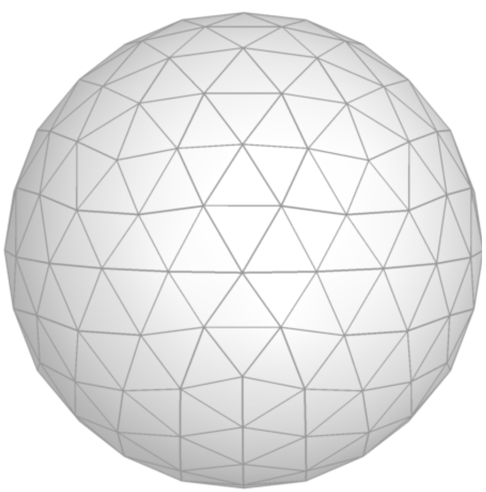
Sphere: Surface Mesh and Strands

- **Spherical grid obtained by repeated subdivision of an icosahedron to any arbitrary level**
- Resulting geodesic grid provides largely isotropic triangular cells over the sphere
- Newly formed points are flushed to the surface of the sphere
- Strands do not intersect
- **Hamiltonian loops in each layer are self similar**

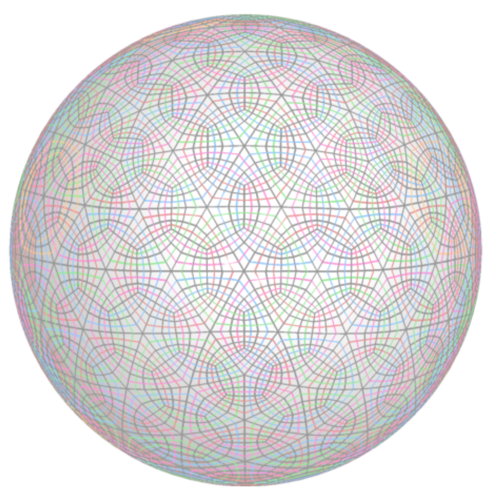


Icosahedron

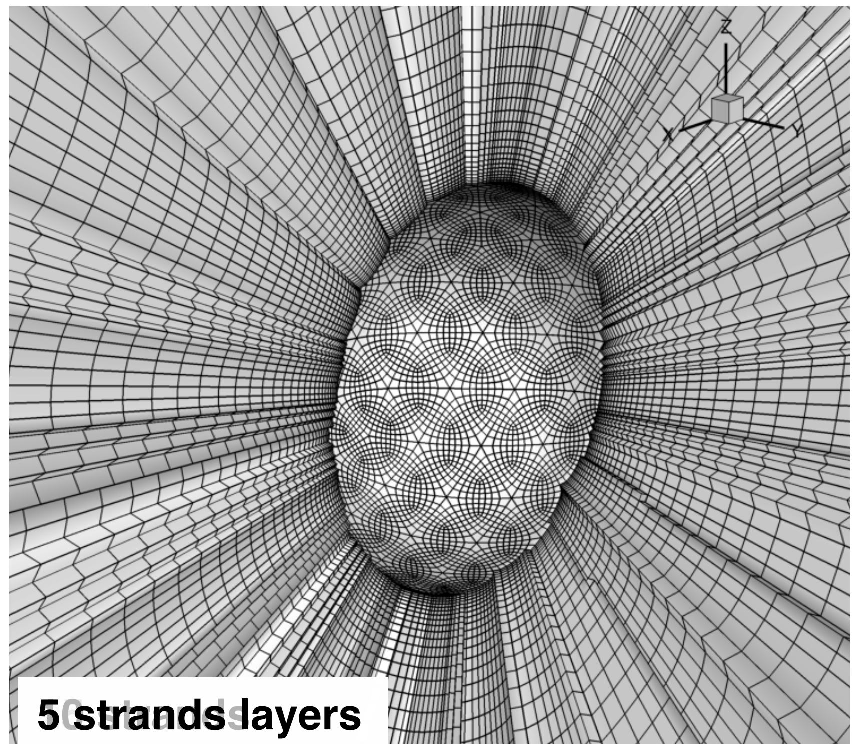
Subdivision process



Surface triangular geodesic grid



Hamiltonian loops on the surface

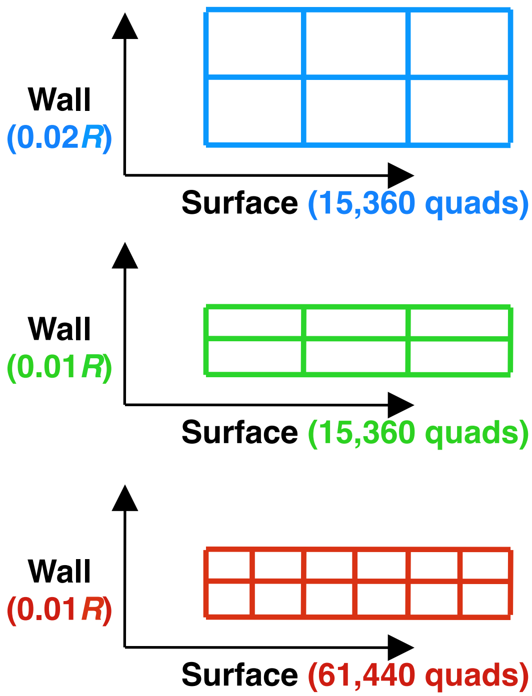


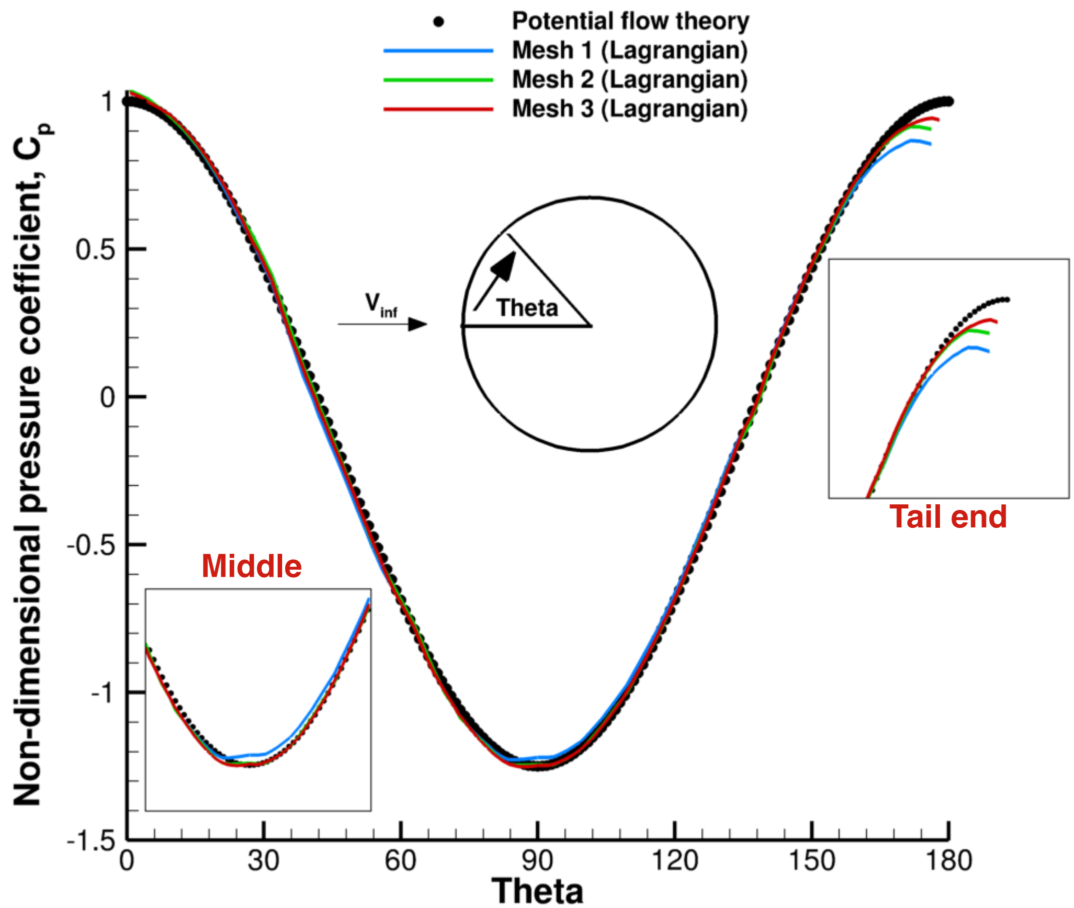
layers

Govindarajan et al. AHS 2015

Inviscid Sphere: Surface Pressure

- Flow at low Mach number over a sphere compared against potential flow theory
- Freestream Mach number of 0.2 using MUSCL reconstruction and DDLGS implicit scheme

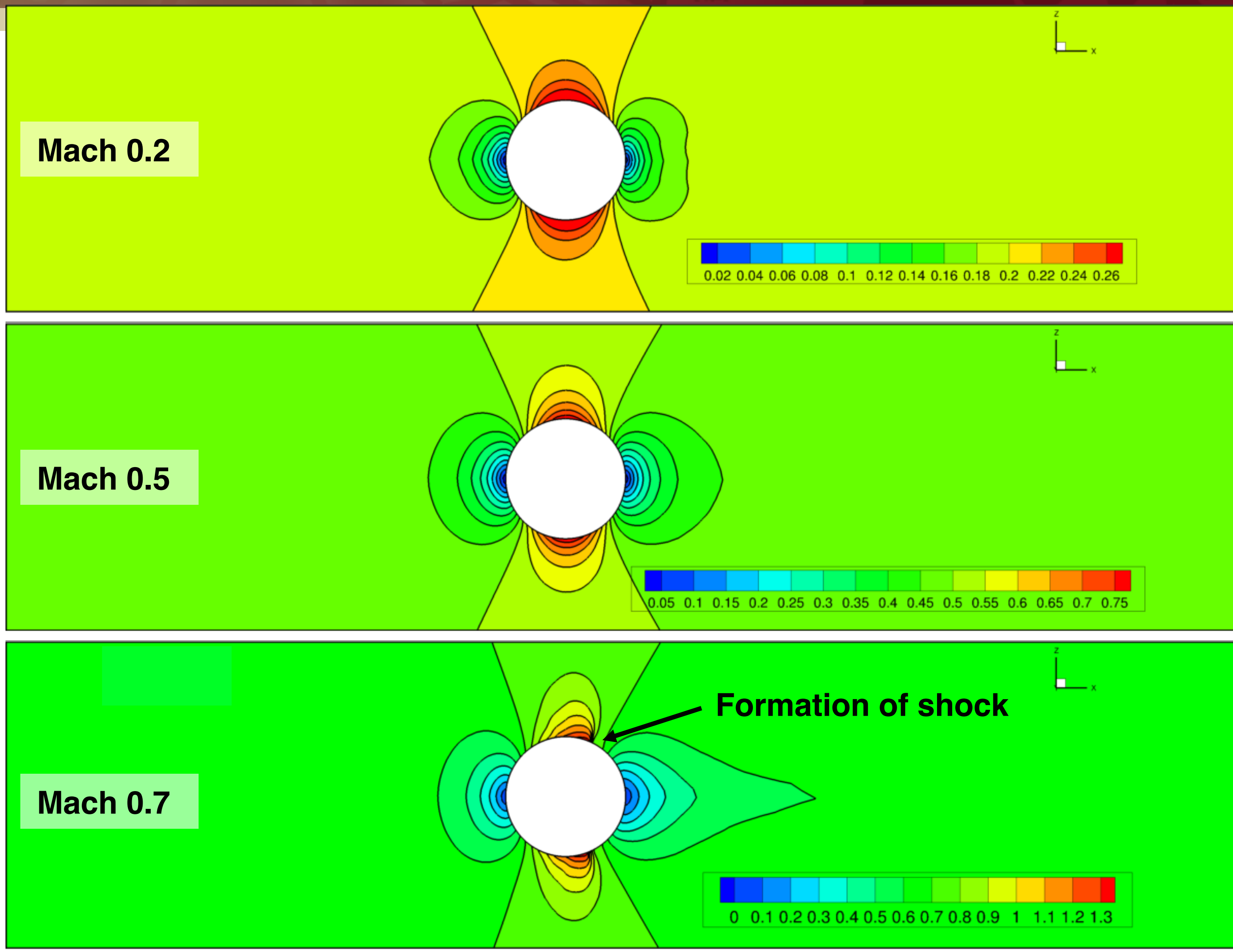




Govindarajan et al. AHS 2015

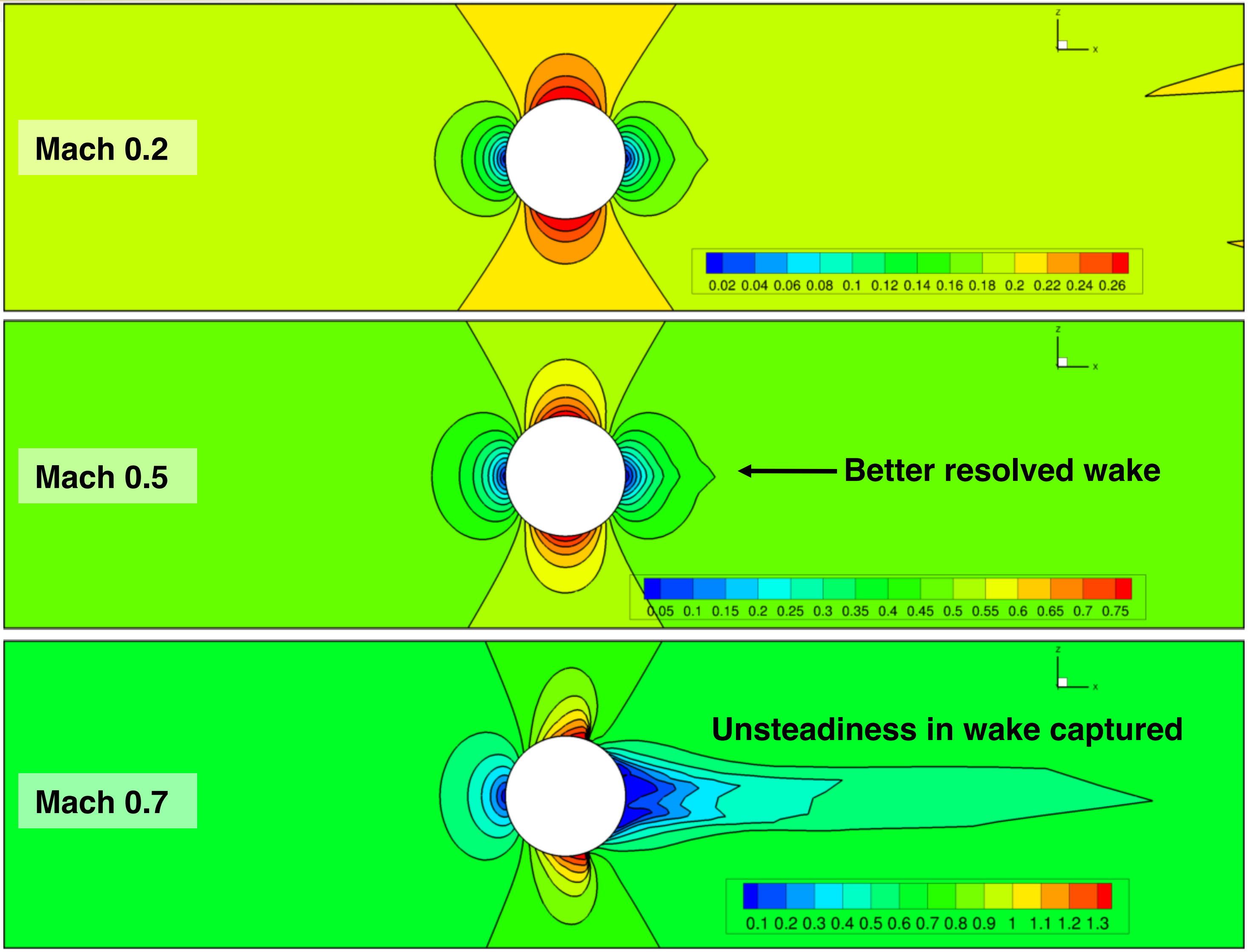
UNCLASSIFIED

Inviscid Flow Over Sphere (1st Order)



- **Flow physics reasonably well captured**

Inviscid Flow Over Sphere (3rd Order)



- **Wake and shock better captured with increased accuracy of reconstruction scheme**

Viscous Flow over Sphere: Streamlines

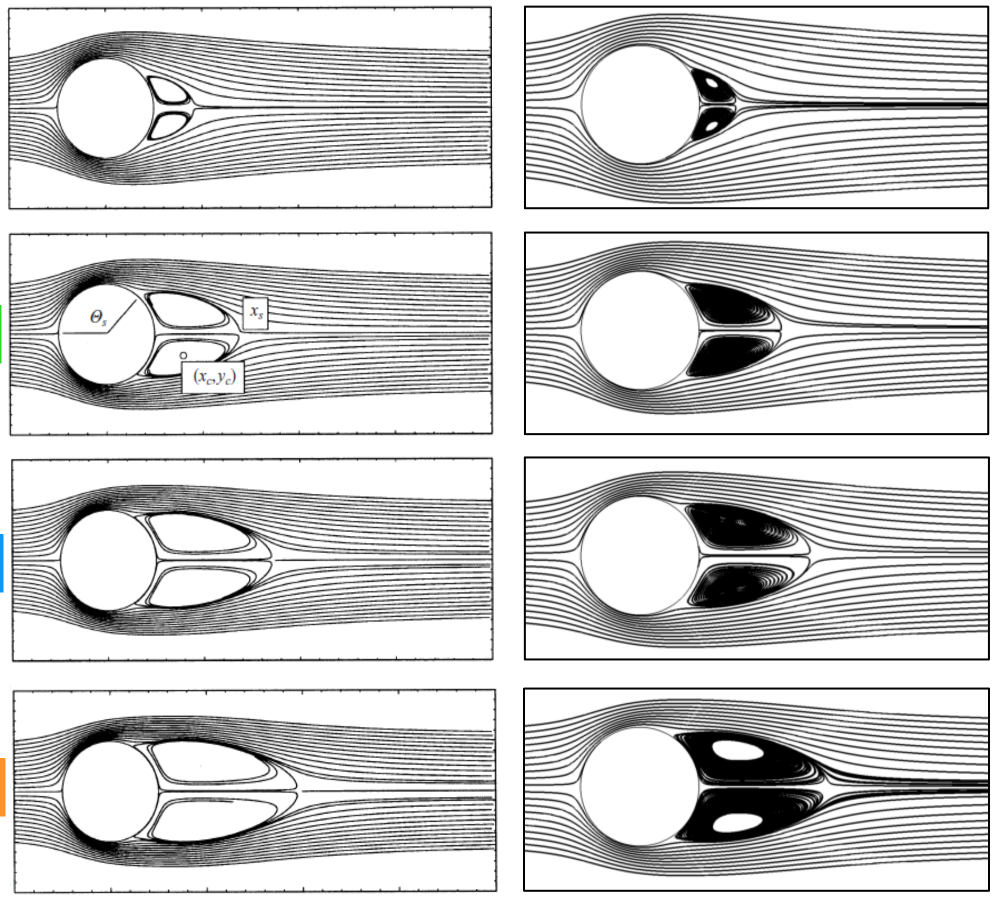
Structured Solver (Reference)* | Hamiltonian/Strand

Re 50

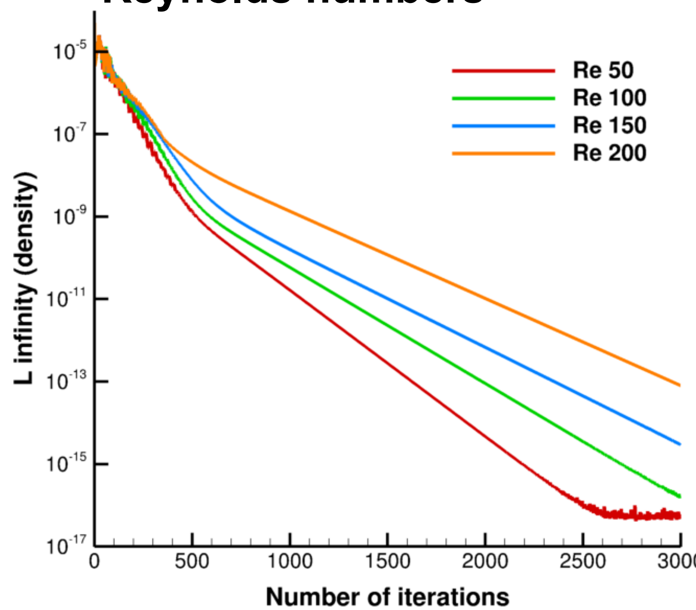
Re 100

Re 150

Re 200



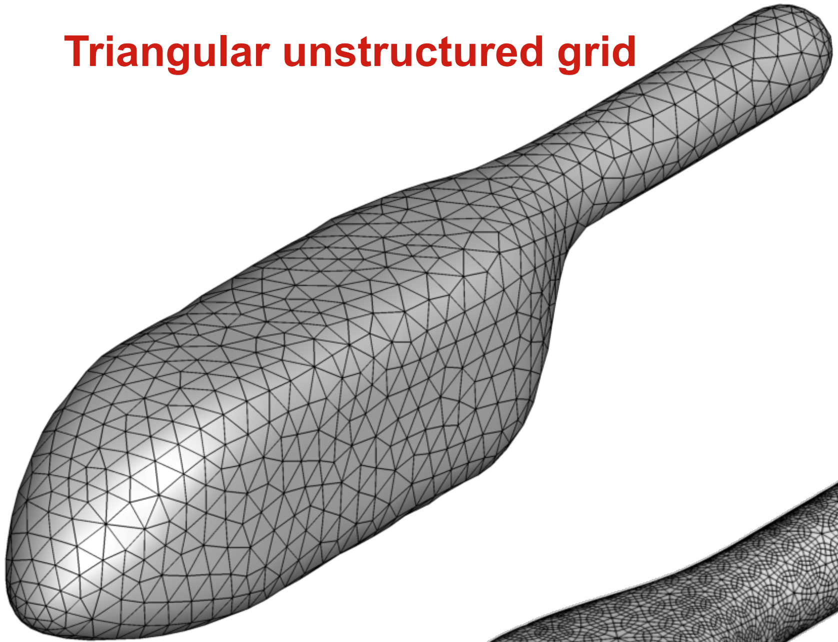
- Viscous flow past sphere compared for low Reynolds numbers
- Size of wake regions increases as Reynolds number increases
- Residual convergence till machine zero for all tested Reynolds numbers



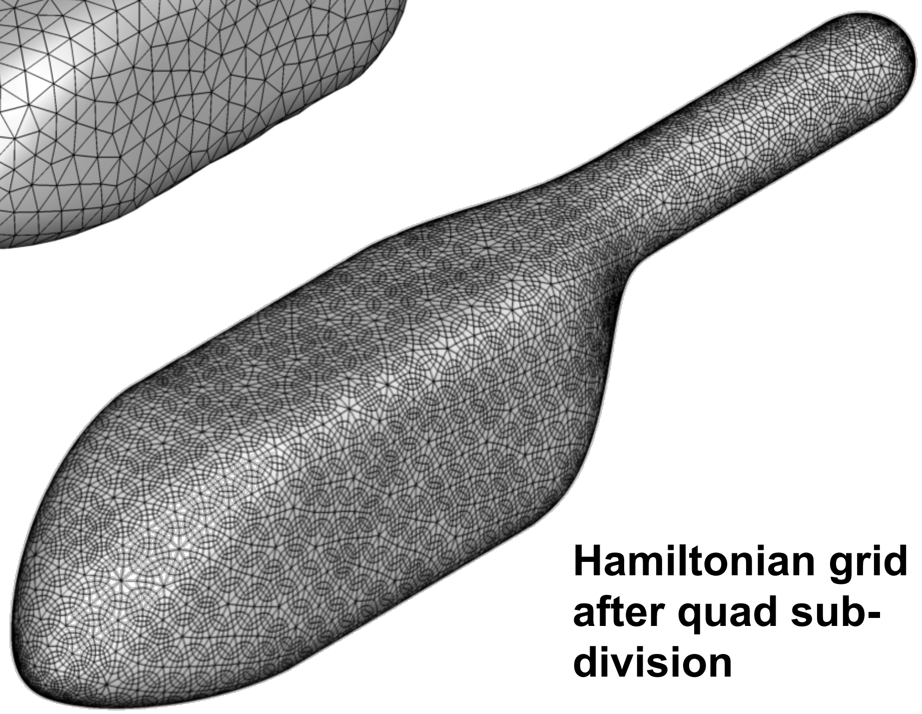
Reference: Flow Past Sphere up to Reynolds Number of 300, T. A. Johnson and V. C. Patel (J. Fluid Mechanics, 1999)

Robin Fuselage: Mesh System

Triangular unstructured grid

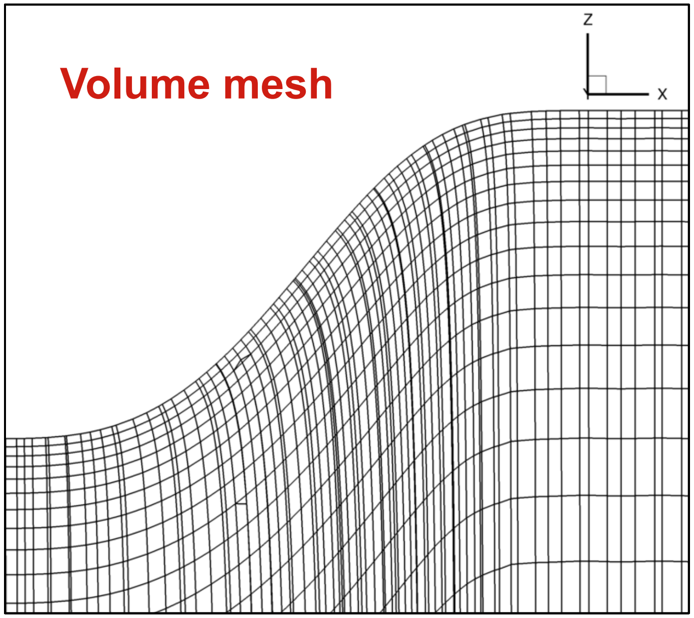


- Representative fuselage shape
- 2,096 triangles, 25,152 quadrilaterals
- Crossover of strands possible
- Advancing front-like technique used to smooth the strand normals



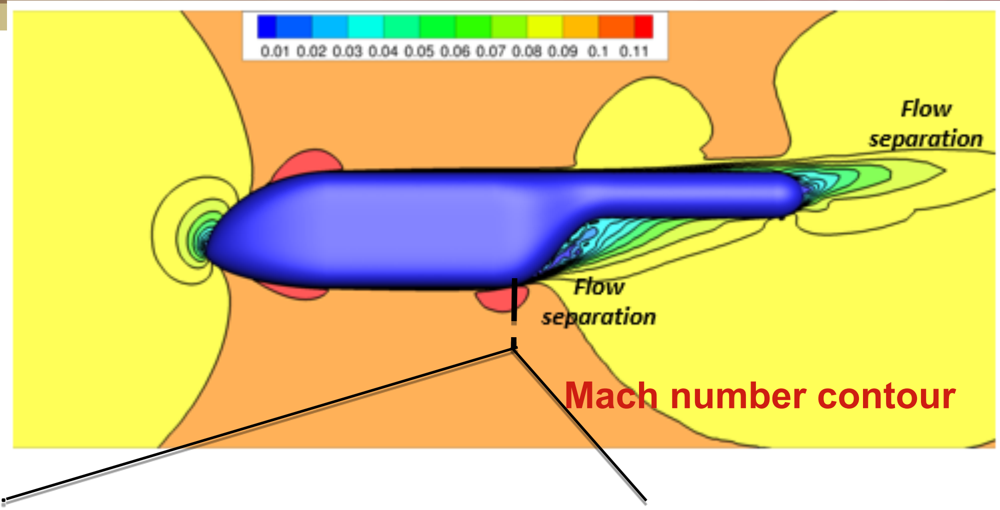
Hamiltonian grid after quad sub-division

Volume mesh



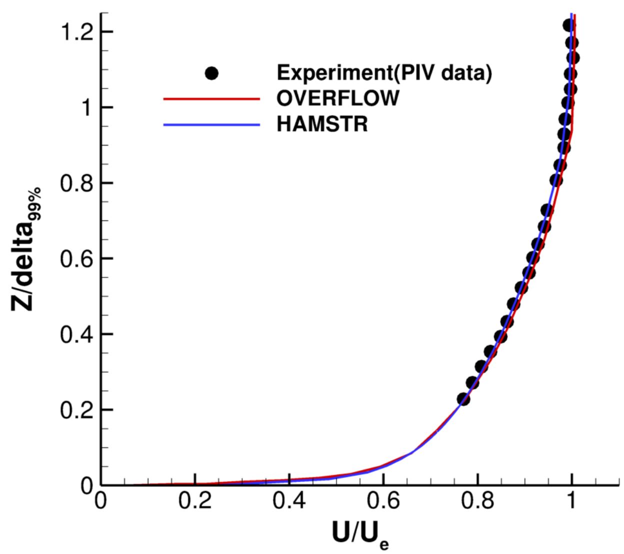
Jung et al, AIAA SciTech 2016

Turbulent Flow: Robin Fuselage

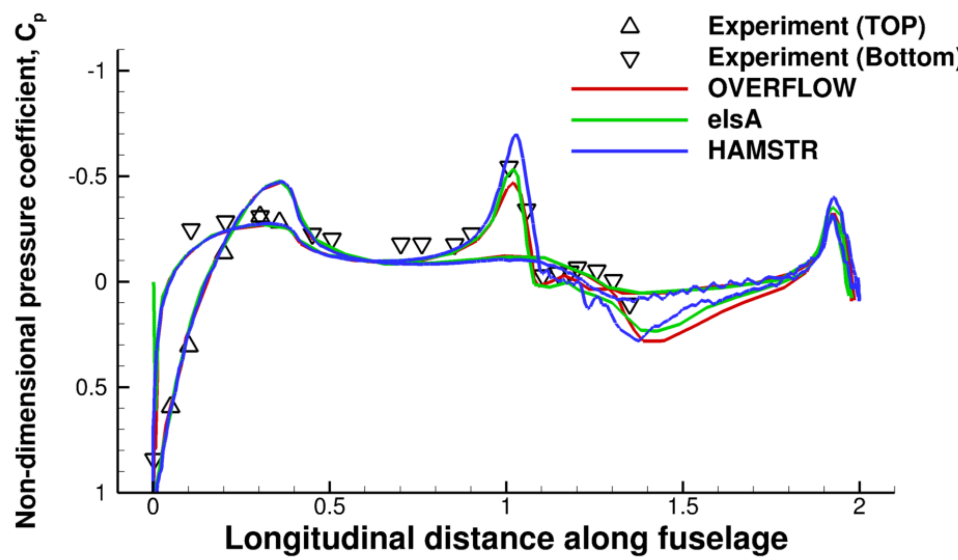


Mach number contour

- Initial spacing 10^{-5}, Cell count 2,000,000
- Mach = 0.1, AoA = 0^{0}, Re = 1,600,000
- Peak surface pressure over-predicted by HAMSTR compared to other solvers
- Wiggles on the aft-end need to be analyzed
- Good agreement in boundary layer predictions on underside of fuselage



Velocity profile in boundary layer



Surface pressure distribution

AIAA SciTech 2016

Questions on viability of the approach

- How would reconstruction schemes such as WENO5 and compact WENO fare on Hamiltonian grids?
- Is the solution quality and convergence strongly dependent on the curvature of Hamiltonian paths? Can we control the curvature?
- Can this be extended to make a functional 3-D RANS solver? With Hamiltonian loops on the surface and strands in the wall normal direction.. Will the results be accurate?
- **Can this approach be parallelized? Will resulting code be scalable?**
- Can the method be used in an overset framework such as HELIOS?
- Can the Hamiltonian loop approach be extended to general surface tesselations?
- How will new points inside cells be introduced such that they flush with the surface (same question as for high-order FE methods)?
- …

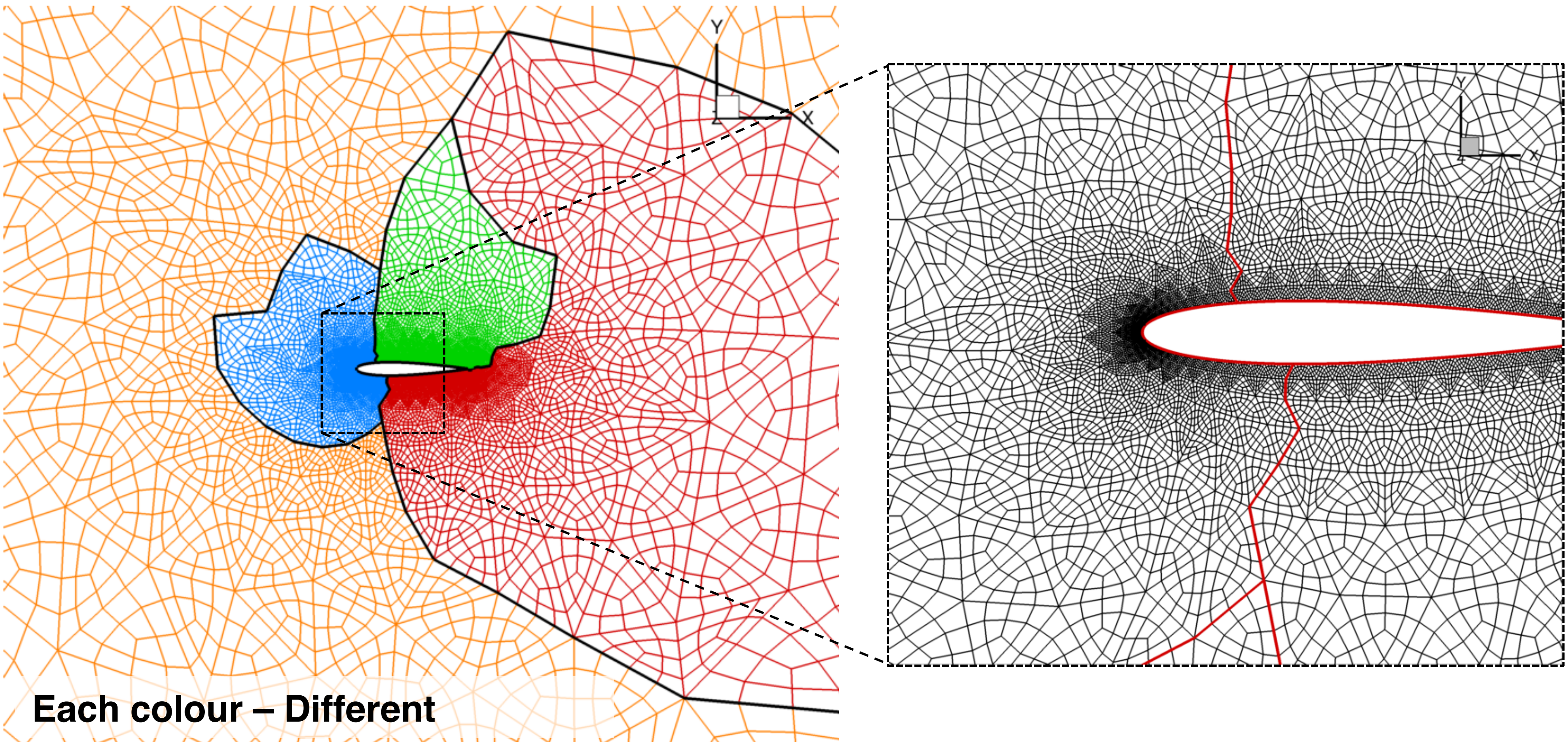
U.S. ARMY RDECOM

AMRDEC

Domain Decomposition

- Domain decomposition performed using Message Passing Interface (MPI)

Partition the triangles on the surface mesh (using METIS) → Create loops for each sub-domain → Create ghost cell information at boundaries between sub-domain → Create strand meshes for each sub-domain



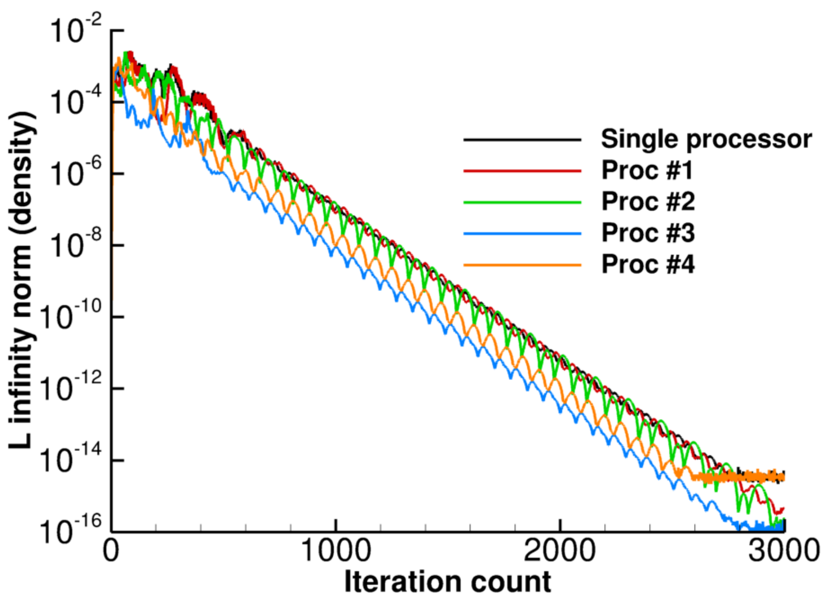
Each colour – Different processor

Govindarajan et al, AHS 2015

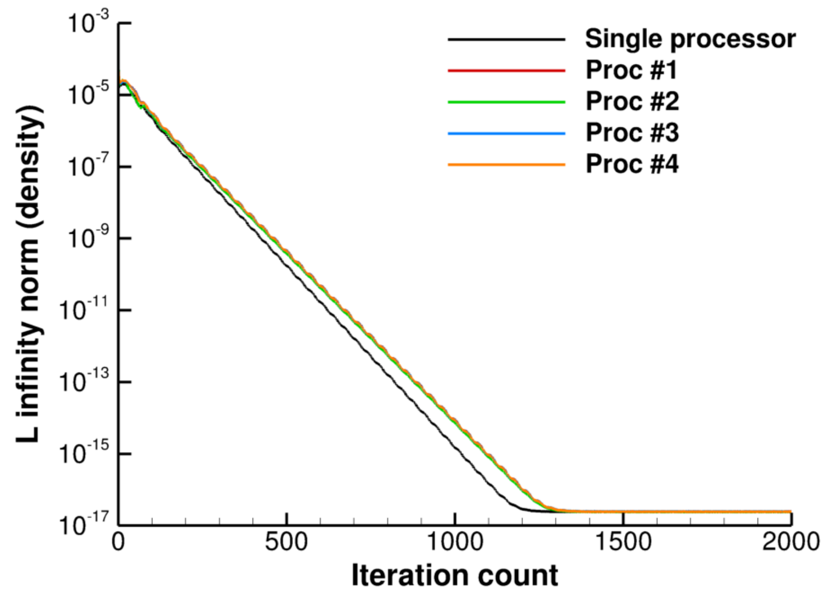
UNCLASSIFIED

TECHNOLOGY DRIVEN. WARFIGHTER FOCUSED.

Domain Decomposition: Convergence



Two-dimension

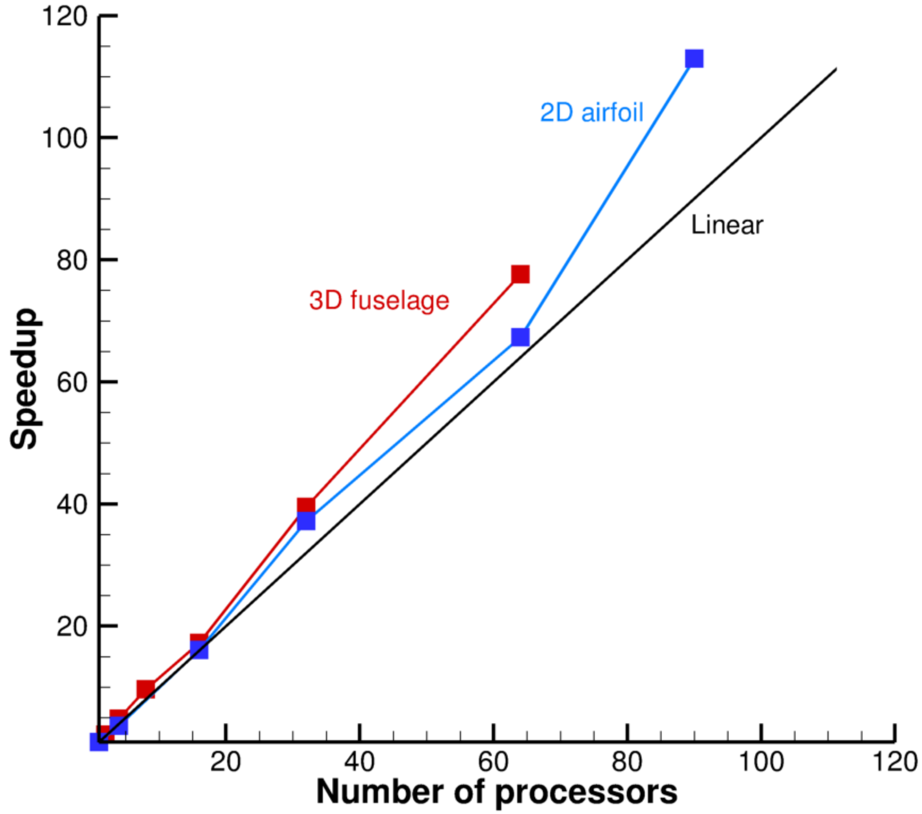


Three-dimension

- **Convergence histories compared between a single processor and multiple processors**
- Third-order **reconstruction and implicit scheme**
- Convergence trend of each processor **nearly linear with the same rate of residual drop**
- Convergence rates **similar between single and multiple processors**

Govindarajan et al., AHS 2015

Domain Decomposition: Scalability



- **Strong scaling tests** **performed on an 2d airfoil and 3d fuselage using third-order reconstruction and implicit inversion on UMD Deepthought II**
- Linear speed-up observed with an increase in number of processors
- Super-linear behaviour maybe attributed to memory fitting into the cache

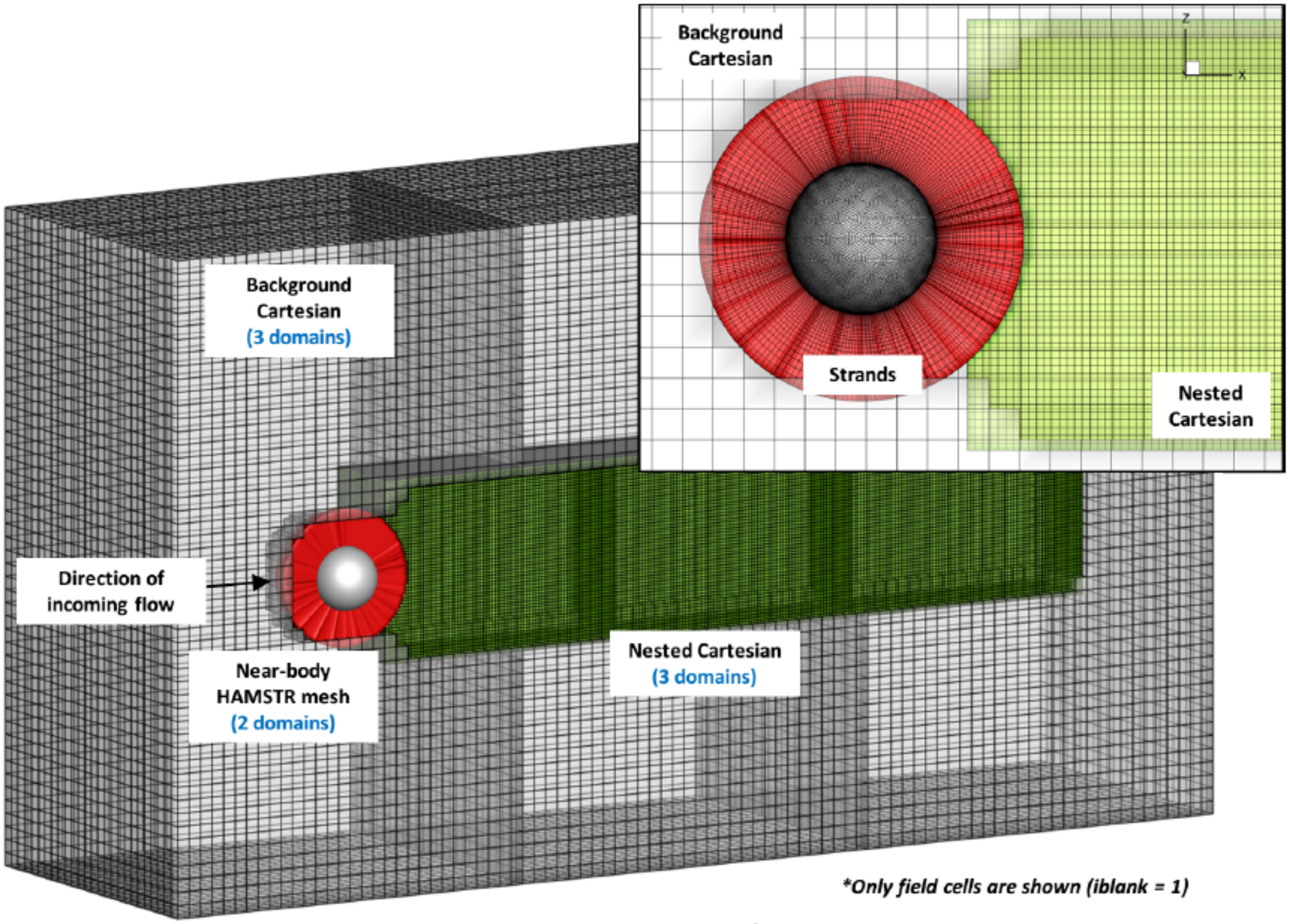
Govindarajan, AHS 2015

Questions on viability of the approach

- How would reconstruction schemes such as WENO5 and compact WENO fare on Hamiltonian grids?
- Is the solution quality and convergence strongly dependent on the curvature of Hamiltonian paths? Can we control the curvature?
- Can this be extended to make a functional 3-D RANS solver? With Hamiltonian loops on the surface and strands in the wall normal direction.. Will the results be accurate?
- Can this approach be parallelized? Will resulting code be scalable?
- Can the method be used in an overset framework such as HELIOS?
- Can the Hamiltonian loop approach be extended to general surface tesselations?
- How will new points inside cells be introduced such that they flush with the surface (same question as for high-order FE methods)?
- …

Overset grid framework

- **Leverage open-source package TIOGA to create an overset grid framework *(https://github.com/jsitaraman/tioga)***

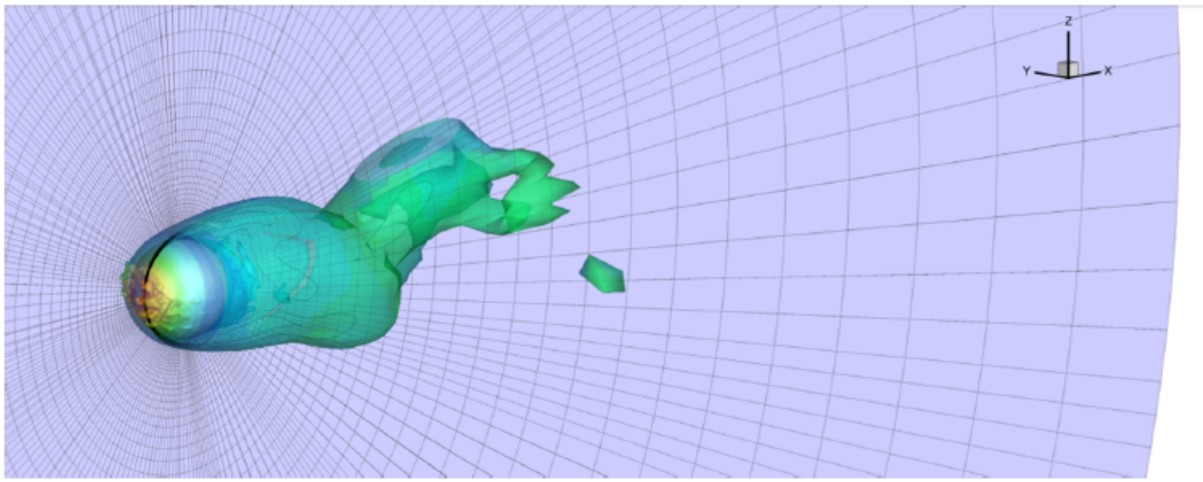


All grids are solved using HAMSTR

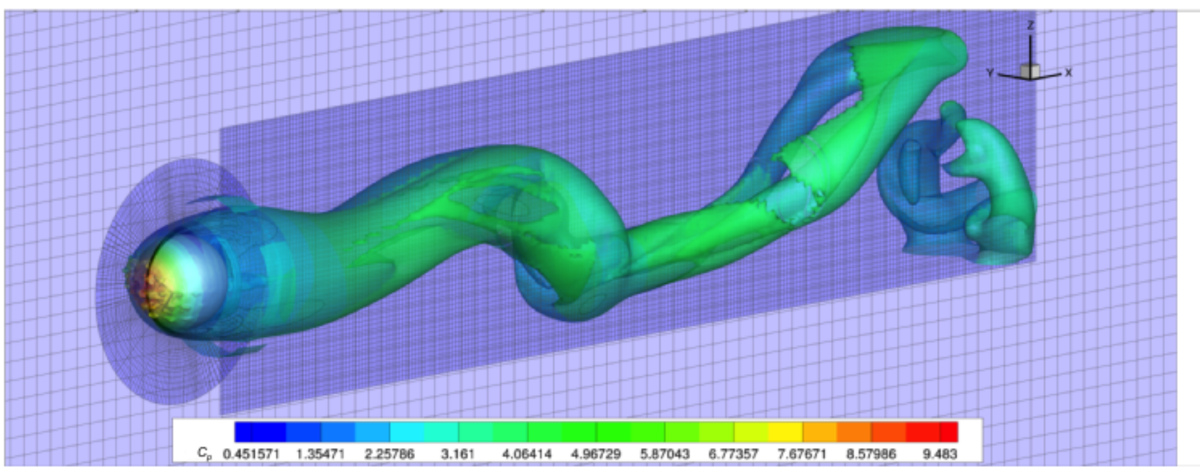
It is easy to make Hamiltonian-path type data structure out of a structured grid

To be presented by Jung at AHS 2016

Flow over a sphere with overset grids



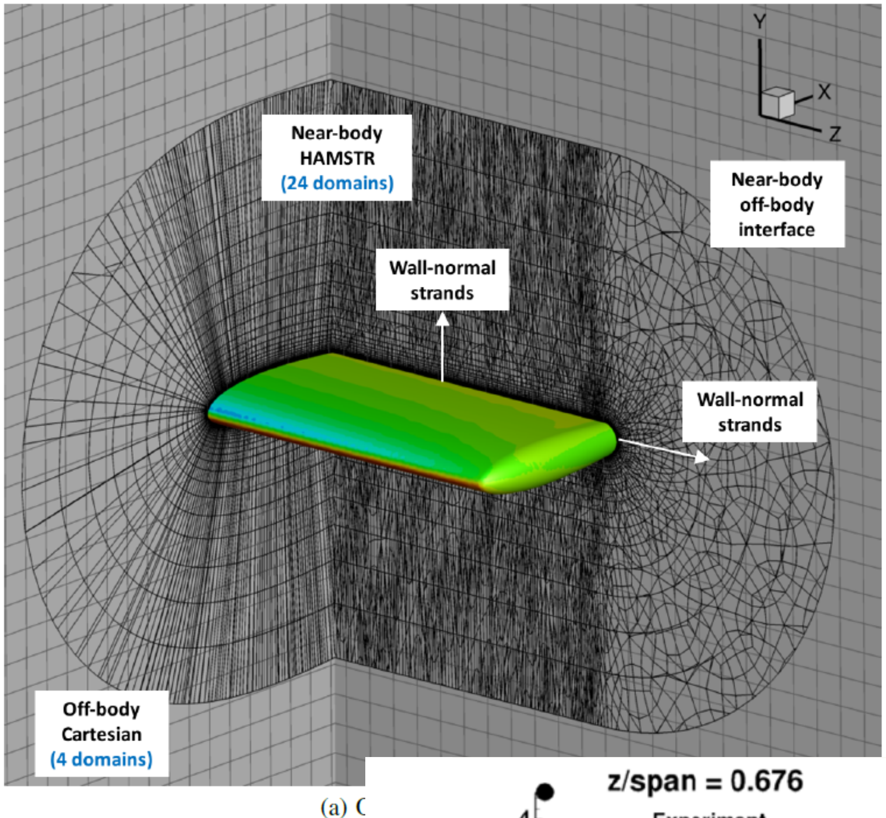
(a) Isolated strand grid

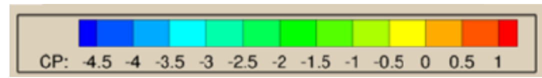


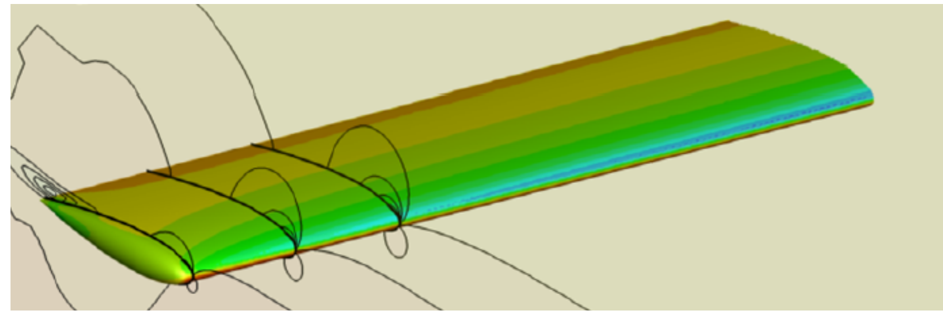
(b) Strand grid and nested Cartesian grid

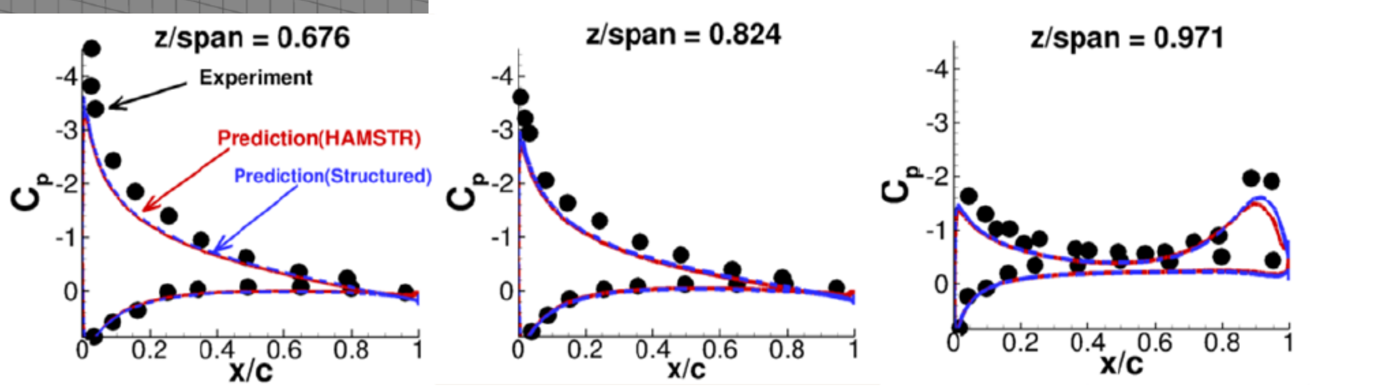
To be presented by Jung at AHS 2016

NACA 0015 RANS solutions
(M=0.1235, aoa=12 deg, Re = 6 mil)



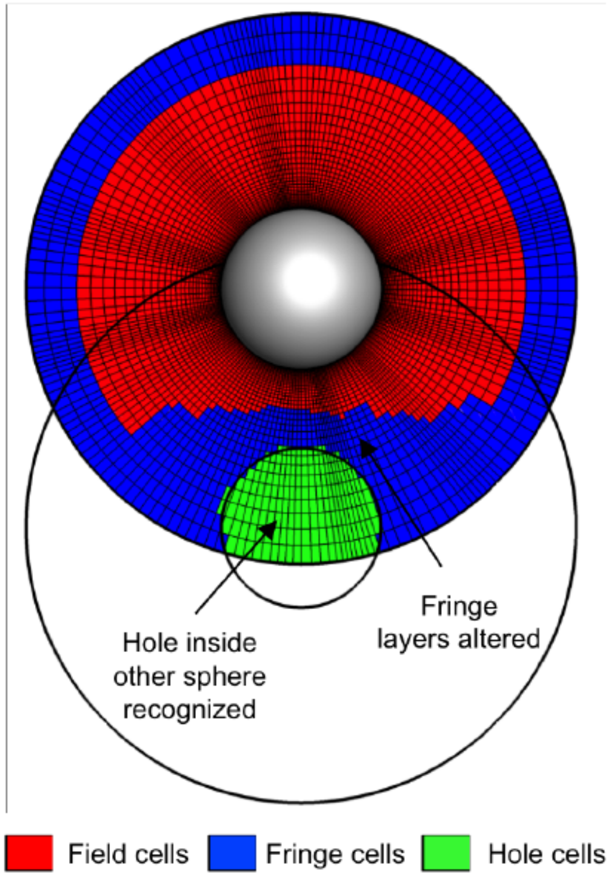




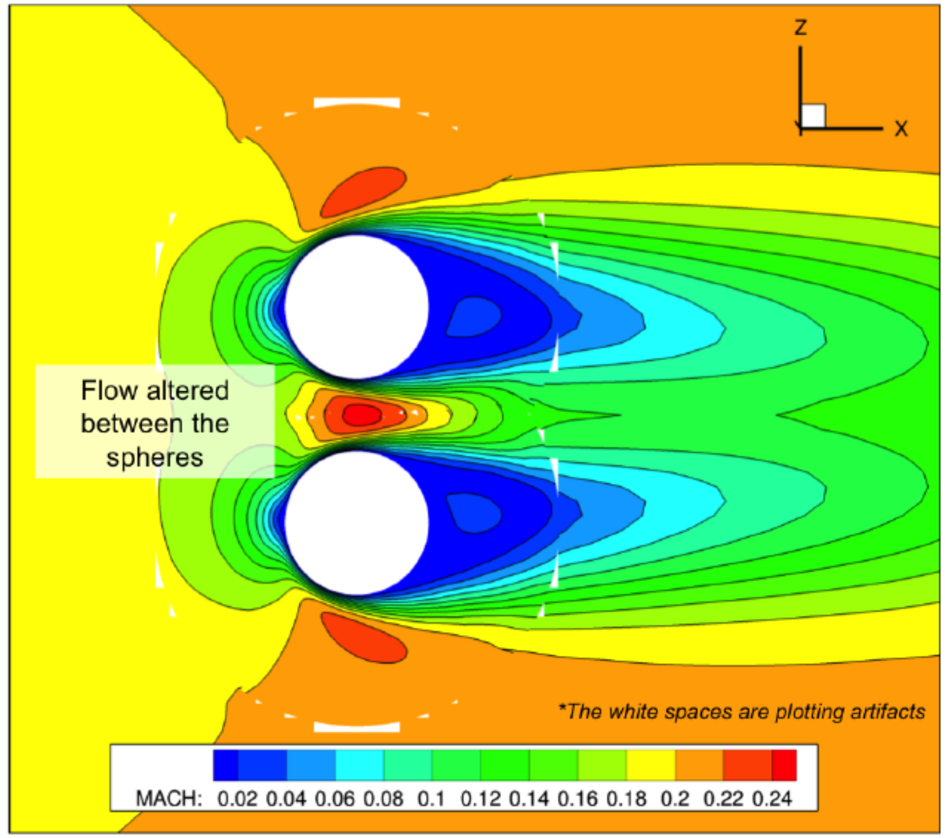


To be presented by Jung et al at AHS 2016

Tandem Spheres (Re=100, M=0.2)



(a) i-blanking map of a sphere

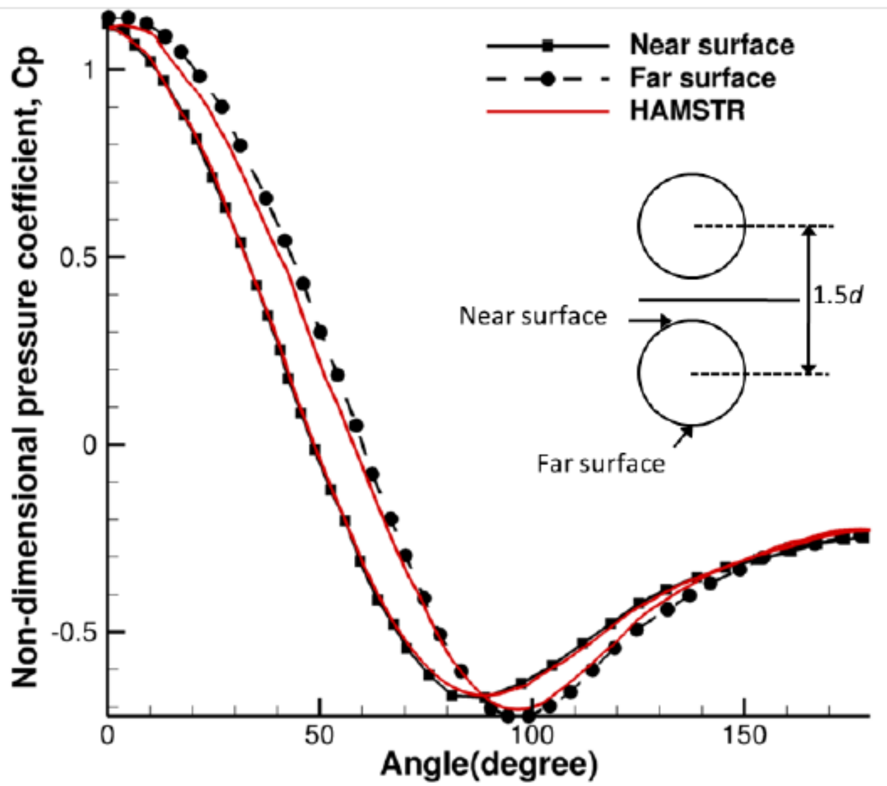


(b) Mach contour lines

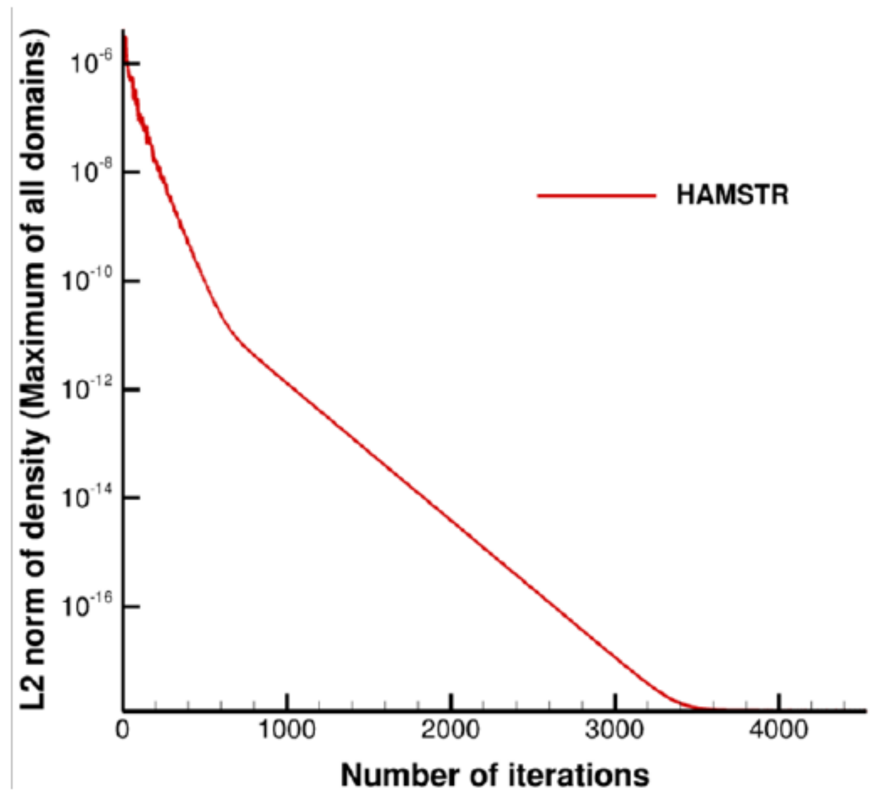
To be presented by Jung et al, at AHS 2016

UNCLASSIFIED

Tandem Sphere validation



(a) Surface pressure distribution



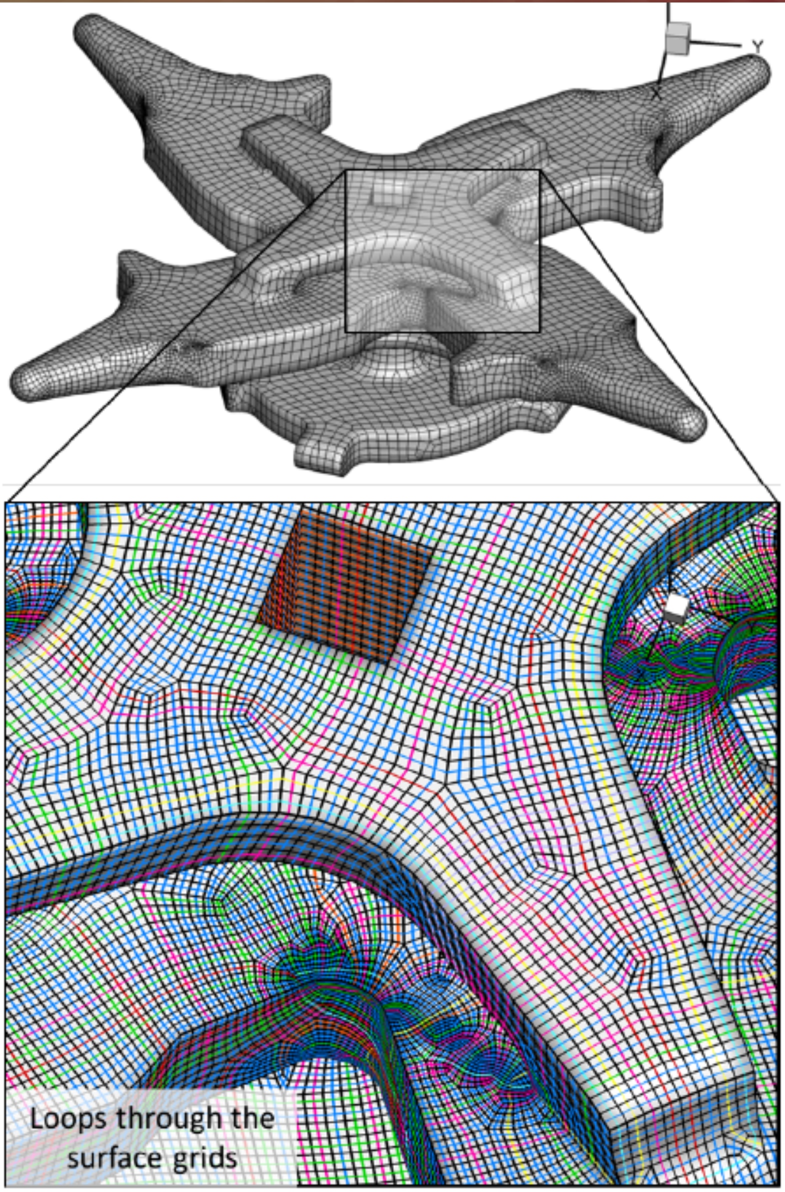
(b) Residual convergence

To be presented by Jung at AHS 2016

Questions on viability of the approach

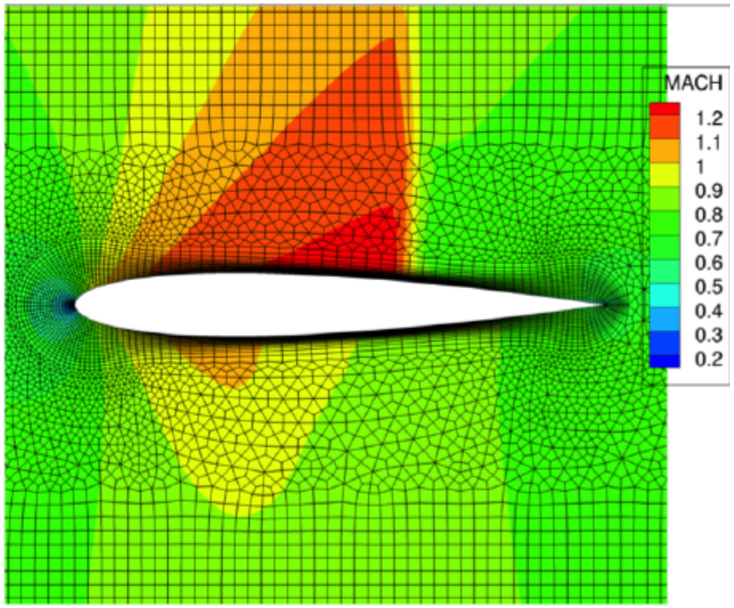
- How would reconstruction schemes such as WENO5 and compact WENO fare on Hamiltonian grids?
- Is the solution quality and convergence strongly dependent on the curvature of Hamiltonian paths? Can we control the curvature?
- Can this be extended to make a functional 3-D RANS solver? With Hamiltonian loops on the surface and strands in the wall normal direction.. Will the results be accurate?
- Can this approach be parallelized? Will resulting code be scalable?
- Can the method be used in an overset framework such as HELIOS?
- **Can the Hamiltonian loop approach be extended to general surface tesselations?**
- How will new points inside cells be introduced such that they flush with the surface (same question as for high-order FE methods)?
- …

General Tesselations



In General mixed element tesselations, Hamiltonian paths can cross themselves (i.e. like a figure eight).

This can potentially affect convergence. However, testing revealed that the solver is robust and does achieve machine-zero convergence even in the presence of self-crossing paths.



To be presented by Jung at AHS 2016

Concluding Remarks

- An idea of forming linelets using quadrilateral sub-division was introduced in 2-D. Stencil based discretization schemes and approximate factorization methods were used to devise efficient and accurate numerical algorithms similar to those typical to structured grids
- The HAMSTR Solver developed at UMD extended and advanced the original Univ of Wyoming work to 3-D. HAMSTR has been validated to be accurate for a range of test cases
- HAMSTR is ready for prime-time now and will be deployed as a near-body solver in U.S Army Helios in the near-future.
- Further developments and advancements in several areas are still necessary to achieve full potential as a production ready solver.

<u>Bottom line</u>

- HAMSTR creates an approach where unstructured prizmatic grids and structured grids can be treated in a unified manner, i.e. the grids are represented as a collection of curves. Unstructured prizmatic grids need sub-division of their original elements, while structured grids (multi-block included) can be used as-is.

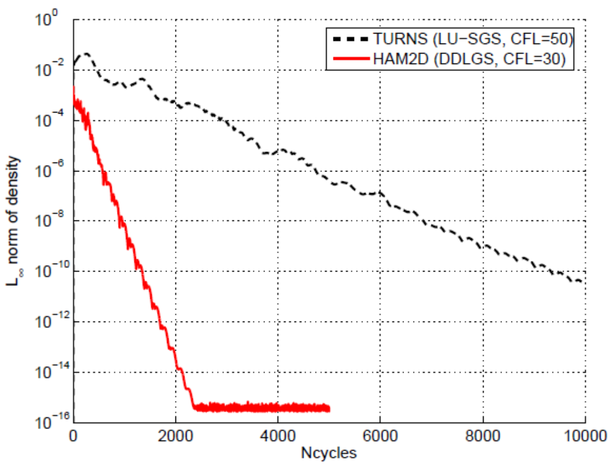
Future work

- Meshing improvements, CAD interface to find interior points flushed to the surface + strand generation issues
- Adapting the grid topology for FE type discretizations
  - sub-divided mesh is all-hex, tensor-product type DG methods can be utilized.
  - "Line-DG" approaches (Persson et al) can further help and one can possibly use block implicit operators that goes across elements using the Hamiltonian paths.
  - Iso-geometric analysis that can use NURBS type representation of the Hamiltonian Path curves.
- Solution scheme improvements in finite-volume context
  - Augmenting the solution reconstruction with unstructured-type gradients on a need-basis based on a smoothness indicator
  - h-multigrid implementation: quadrilateral sub-division provides a natural coarse grid sequence
  - GMRES based linear solver that uses all of the approximate operators as preconditioners.

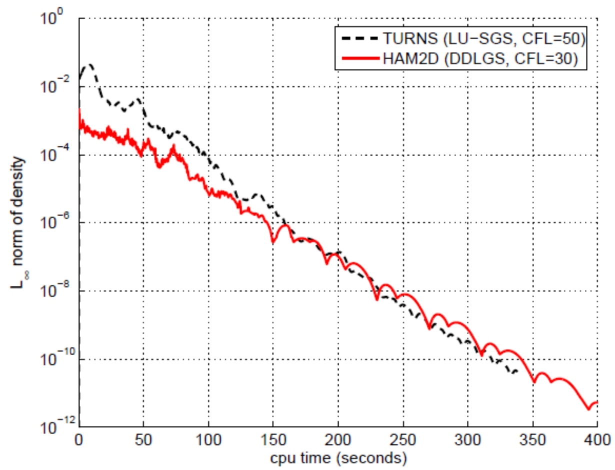
Acknowledgements

- Bharath Govindarajan, Yong Su Jung and Jim Baeder at the University of Maryland. Most of this presentation is their work.
- Beatrice Roget for writing the mesh generator and providing general enthusiasm for this work at Univ of Wyoming even when it was unfunded.
- Bob Meakin at CREATE A/V for believing in the potential of this idea and funding the extension of the original 2-D work.
- Roger Strawn, Andy Wissink and rest of the Helios team for their support and encouragement to explore an unconventional path.

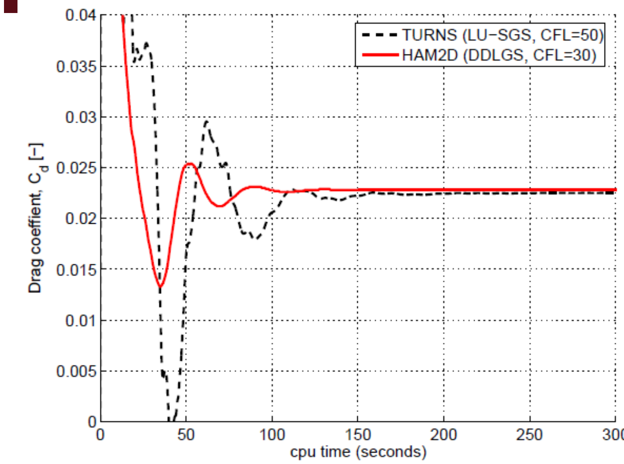
Transonic Airfoil: Efficiency Comparison



Number of iterations



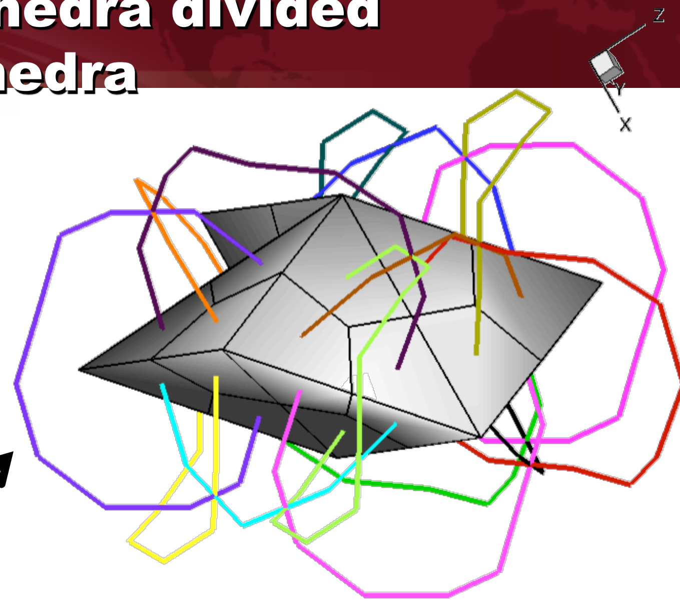
Wall clock time



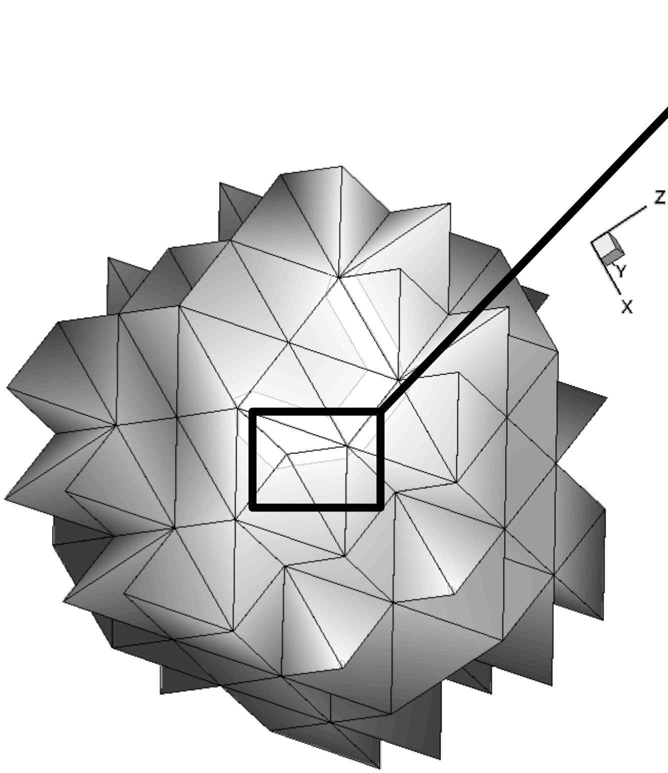
Drag convergence

- **Convergence characteristics compared with the structured TURNS code**
- **Both meshes had ~20k cells**
- **LU-SGS scheme in TURNS is faster on a per-iteration basis (uses spectral radius approximation) , but requires larger number of steps to achieve similar convergence**
- **Drag convergence within 1 count: 150*s* (Hamiltonian), 300*s* (TURNS)**
- **Unstructured Hamiltonian code converges approximately as fast as the structured code**

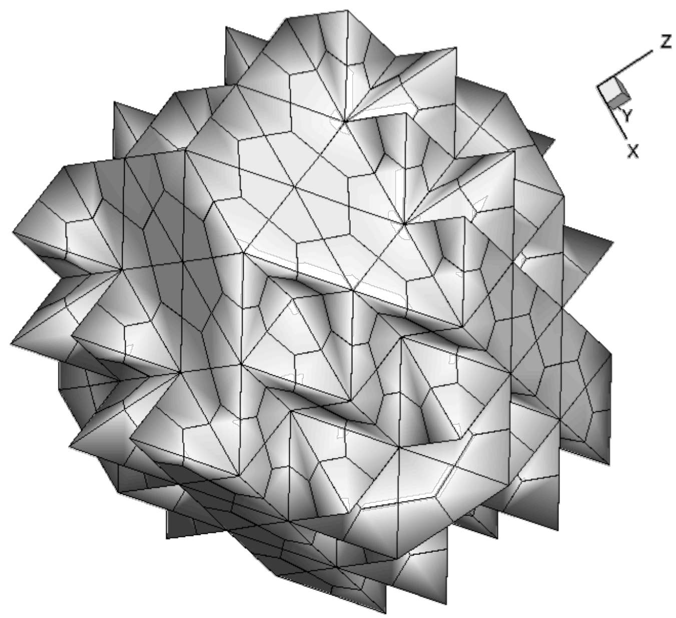
3-D Hamiltonian Chains with tetrahedra divided into hexahedra



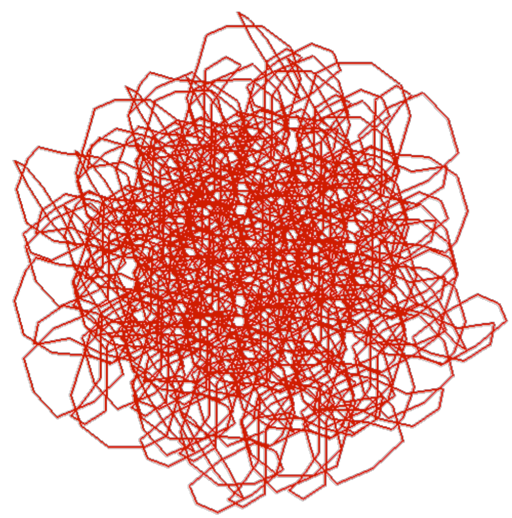
Detail of loops around 6 tetrahedral cells



Tetrahedral mesh-block

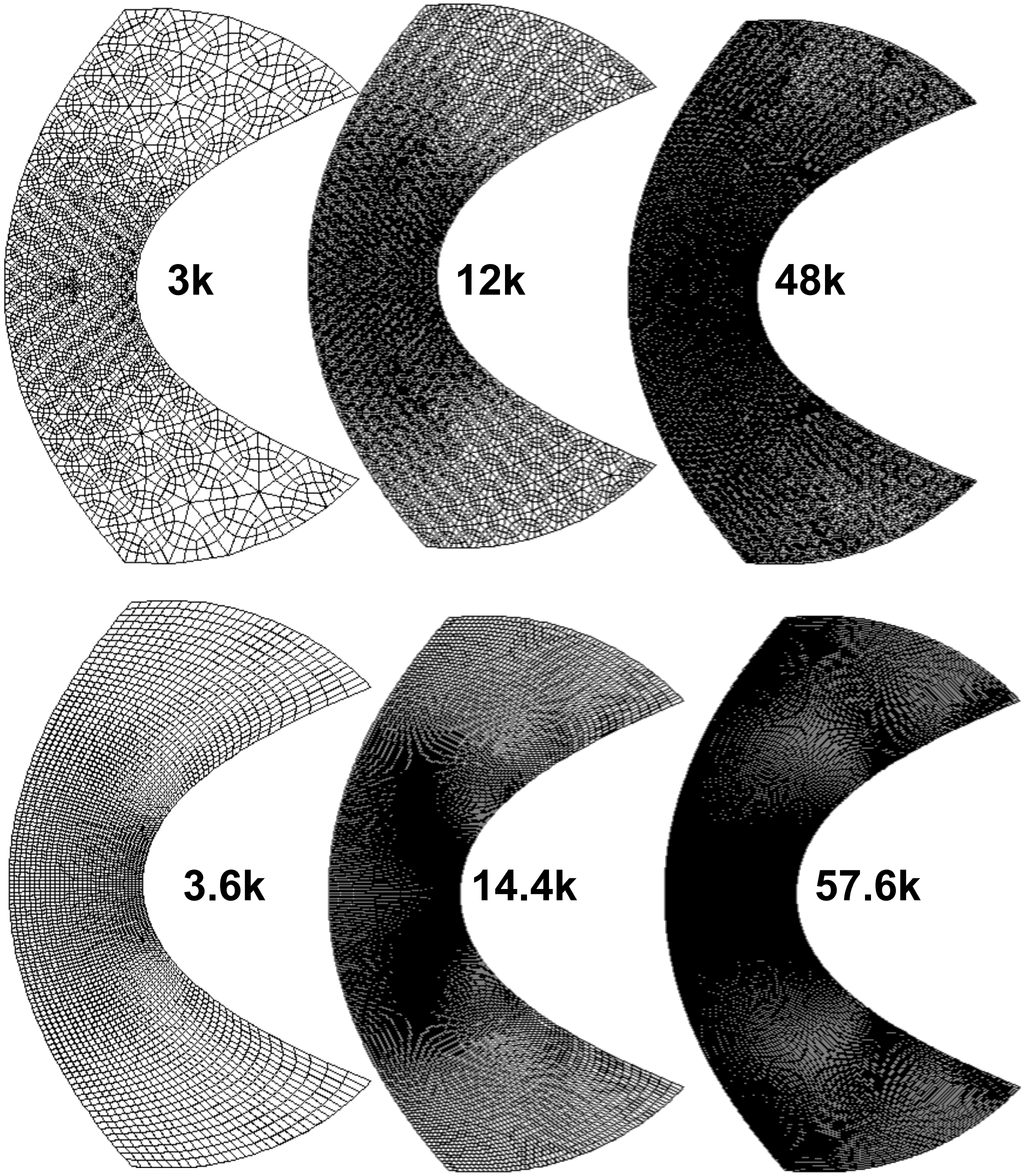


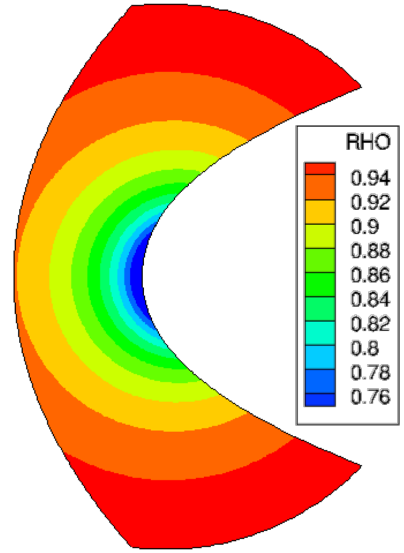
Tetrahedra divided into hexahedra



Loops within mesh-block

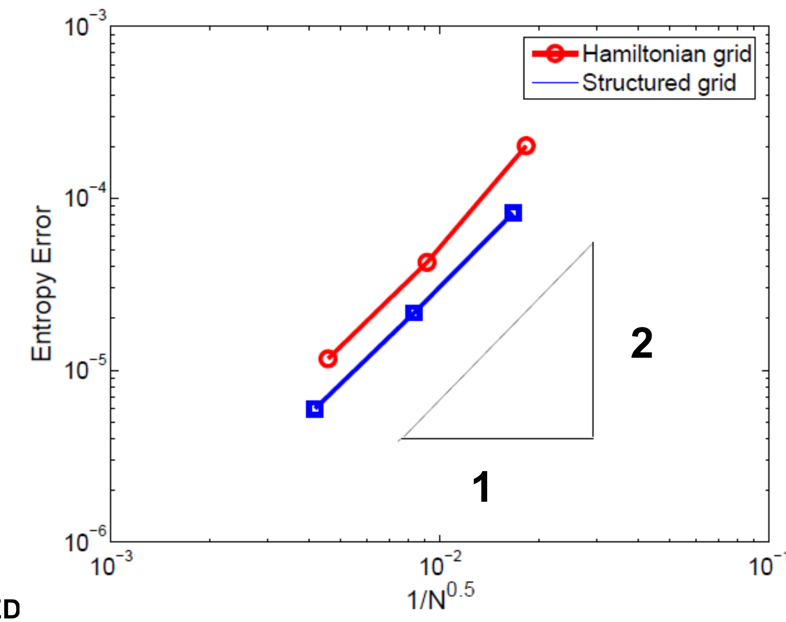
Results : Ringleb flow





Exact solution to Euler equations using Hodograph transformation

Need to generate grids to align with streamlines



AMRDEC Web Site
www.amrdec.army.mil

Facebook
www.facebook.com/rdecom.amrdec

YouTube
www.youtube.com/user/AMRDEC

Public Affairs
AMRDEC-PAO@amrdec.army.mil